



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Diseño de un sistema de control de inversor para un servo  
senoidal de 200W

Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

AUTOR/A: Poveda Serrano, Antonio

Tutor/a: Martínez Román, Javier Andrés

CURSO ACADÉMICO: 2021/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUOLA TÉCNICA  
SUPERIOR INGENIERÍA  
INDUSTRIAL VALENCIA

Curso Académico:





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIERÍA  
INDUSTRIAL VALENCIA

# ÍNDICE DE DOCUMENTOS

---

**MEMORIA**

**PRESUPUESTO**

---



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIERÍA  
INDUSTRIAL VALENCIA

# DISEÑO DE UN SISTEMA DE CONTROL DE INVERSOR PARA UN SERVO SENOIDAL DE 200W

---

# MEMORIA

---

Trabajo Final de Grado

Antonio Poveda Serrano

Julio, 2022



“A mi familia por el apoyo recibido,  
A mi tutor, por el tiempo dedicado,  
A mis compañeros por estar a mi lado.  
...”





## **RESUMEN**

Los motores síncronos de imanes permanentes se utilizan extensivamente en la industria tanto para el posicionamiento, como para robótica, como para tracción ya sea industrial, naval o automovilística.

Esta tecnología está ampliamente utilizada en cualquier área tecnológica relacionada con la industria y su utilización en el campo de la automoción continúa tomando cada día más importancia. Las ventajas de estos motores frente a sus competidores directos se pueden resumir en un mejor rendimiento, mayor densidad de potencia y temperatura de trabajo más reducida. Estos factores los hacen un componente esencial de la tecnología de tracción y posicionamiento.

En el presente Trabajo Final de Grado se llevará a cabo el diseño, implementación y validación de un sistema de control de velocidad y corriente para un motor síncrono de imanes permanentes.

Para esta tarea se hará uso de un microcontrolador de uso popular con capacidades de comunicación inalámbrica para la gestión del control junto con un inversor trifásico para prototipado, que gestionará la aportación de la potencia suficiente a la máquina para poder funcionar.

Finalmente se realizará un diseño de un circuito impreso para su posible fabricación por cualquier fabricante de PCB mediante la generación de los planos de diseño pertinentes y la lista de componentes y su ubicación, incluyendo adicionalmente la lista de materiales necesaria para que cualquier empresa de montaje de tarjetas electrónicas pudiera gestionar la fabricación de las mismas.

**Palabras Clave:** Motor síncrono de imanes permanentes, microcontrolador ESP32, control de corriente y velocidad, diseño de circuito impreso.

## **RESUM**

Els motors síncrons d'imants permanents s'utilitzen extensivament a la indústria tant per al posicionament, com per a robòtica, com per a tracció ja sigui industrial, naval o automobilística.

Aquesta tecnologia està àmpliament utilitzada en qualsevol àrea tecnològica relacionada amb la indústria i la seva utilització al camp de l'automoció continua prenent cada dia més importància. Els avantatges d'aquests motors davant dels seus competidors directes es poden resumir en un millor rendiment, més densitat de potència i temperatura de treball més reduïda. Aquests factors els fan un component essencial de la tecnologia de tracció i posicionament.

En aquest Treball Final de Grau es durà a terme el disseny, la implementació i la validació d'un sistema de control de velocitat i corrent per a un motor síncron d'imants permanents.

Per a aquesta tasca es farà ús d'un microcontrolador d'ús popular amb capacitats de comunicació sense fil per a la gestió del control juntament amb un inversor trifàsic per a prototipat, que gestionarà l'aportació de prou potència a la màquina per poder funcionar.

Finalment es realitzarà un disseny d'un circuit imprès per a la seva possible fabricació per qualsevol fabricant de PCB mitjançant la generació dels plànols de disseny pertinents i la llista de components i la seva ubicació, incloent addicionalment la llista de materials necessària perquè qualsevol empresa de muntatge de targetes electròniques pogués gestionar la fabricació de les mateixes.

**Paraules Clau:** Motor síncron d'imants permanents, microcontrolador ESP32, control de corrent i velocitat, disseny de circuit imprès.

## **ABSTRACT**

Permanent magnet synchronous motors are used extensively in industry for both positioning, robotics, and industrial, naval or automotive traction.

This technology is widely used in any technological area related to the industry and its use in the automotive field continues to gain more importance every day. The advantages of these motors compared to their direct competitors can be summed up in better performance, higher power density and lower operating temperature. These factors make them an essential component of traction and positioning technology.

In this Final Degree Project, the design, implementation and validation of a speed and current control system for a permanent magnet synchronous motor will be carried out.

For this task, a popular microcontroller with wireless communication capabilities will be used for control management together with a three-phase inverter for prototyping, which will manage the supply of sufficient power to the machine to be able to function.

Finally, a printed circuit board design will be made for its possible manufacture by any PCB manufacturer by generating the pertinent design plans and the list of components and their location, additionally including the list of materials necessary for any PCB assembly company. electronic cards could manage their manufacture.

**Keywords:** Permanent magnet synchronous motor, ESP32 microcontroller, current and speed control, printed circuit design.

# **ÍNDICE DE LA MEMORIA**

1.	INTRODUCCIÓN .....	1
1.1.	OBJETIVOS .....	2
1.2.	ANTECEDENTES .....	2
1.3.	ESTRUCTURA DE LA MEMORIA .....	3
2.	DISEÑO DE LA REALIMENTACIÓN DE POSICIÓN.....	5
2.1.	Introducción .....	5
2.2.	Descripción de las señales existentes .....	5
2.3.	DESCRIPCIÓN DE SEÑALES PRODUCIDAS POR EL SERVOPACK .....	6
2.4.	FILTRADO PASO BAJO Y DIVISOR RESISTIVO .....	9
2.5.	OPTOACOPLADOR CON SALIDA TRANSISTOR .....	10
2.4.1.	SELECCIÓN DEL NÚMERO DE PULSOS POR REVOLUCIÓN DEL ENCODER INTERNO AL SERVOPACK .....	12
2.4.2.	SEÑALES CON OPTOACOPLADOR.....	14
2.4.3.	CÓDIGO DE MEDIDA DE POSICIÓN Y PRIMERAS PRUEBAS .....	16
2.4.4.	DETECCIÓN DEL SENTIDO DE GIRO Y ÁNGULO .....	18
2.4.5.	ENSAYO DE TENSIONES DE FASE .....	22
2.6.	REDISEÑO DE LOS OPTOACOPLADORES.....	29
3.	DISEÑO DEL SISTEMA DE CONTROL .....	35
3.1.	INTRODUCCIÓN .....	35
3.2.	ANÁLISIS DE UN MOTOR SÍNCRONO DE IMANES PERMANENTES.....	35
3.3.	CONTROL DEL MOTOR DE IMANES PERMANENTES .....	39
3.4.	ECUACIONES DEL MOTOR DE IMANES PERMANENTES .....	44
3.5.	MODELADO DE LA MÁQUINA CON SIMULINK .....	49
3.6.	AJUSTE DEL CONTROL DE CORRIENTE.....	53
3.7.	IMPLEMENTACIÓN EN SIMULINK DEL CONTROL DE VELOCIDAD .....	55
3.8.	IMPLEMENTACIÓN TEÓRICA DE LA MODULACIÓN DE VECTOR ESPACIAL .....	61
3.9.	DESCRIPCIÓN DEL PROTOTIPO .....	69
3.10.	SENSOR DE CORRIENTE .....	71
3.11.	CALIBRACIÓN DE LA ORIENTACIÓN DEL CAMPO DEL ROTOR.....	77
3.12.	CONTROLADOR DE CORRIENTE.....	80
3.13.	OBTENCIÓN DE LA LECTURA DE VELOCIDAD DEL EJE .....	88
3.14.	CONTROL DE VELOCIDAD .....	89
4.	DISEÑO DE LA TARJETA DE CONTROL PARA FABRICACIÓN EN CIRCUITO IMPRESO.....	94
4.1.	INTRODUCCIÓN .....	94

4.2.	TECNOLOGÍA DE FABRICACIÓN DE LA TARJETA DE CONTROL .....	94
4.3.	STACK UP DE LA TARJETA .....	95
5.	CONCLUSIONES .....	97
6.	Referencias .....	99

## **ÍNDICE DE FIGURAS**

<i>Figura 1 Descripción funcionamiento encoder interno (Yaskawa, <math>\Sigma</math>-II Series SGMH User Manual, 2003)</i>	<i>5</i>
<i>Figura 2 Esquema de funcionamiento de un encoder óptico (AKM)</i>	<i>6</i>
<i>Figura 3 Oscilograma de la señal del encoder sin giro y servo encendido (elaboración propia)</i>	<i>6</i>
<i>Figura 4 Oscilograma de la señal del encoder con giro y servo encendido (elaboración propia)</i>	<i>7</i>
<i>Figura 5 Oscilograma de la señal A con giro y servo encendido con referencia a la señal /A (elaboración propia)</i>	<i>7</i>
<i>Figura 6 Oscilograma de la señal A con giro manual y referencia a masa (elaboración propia)</i>	<i>8</i>
<i>Figura 7 Oscilograma de la señal A con giro y encoder encendido tras el filtrado (elaboración propia)</i>	<i>9</i>
<i>Figura 8 Diagrama de conexiones de las salidas del encoder del servopack (Yaskawa, <math>\Sigma</math>-II Series SGMH User Manual, 2003)</i>	<i>10</i>
<i>Figura 9 Esquema de conexión del circuito de acoplamiento con optoacoplador con salida transistor (elaboración propia)</i>	<i>11</i>
<i>Figura 10 Esquema de conexión recomendada por el fabricante de optoacopladores (LiteON)</i>	<i>12</i>
<i>Figura 11 Esquema de tiempos de la señal a través del optoacoplador (elaboración propia)</i>	<i>13</i>
<i>Figura 12 Gráfica de Voltaje-Corriente de la salida transistor del optoacoplador (LiteON)</i>	<i>13</i>
<i>Figura 13 Oscilograma de la señal A con giro y encoder encendido a través del optoacoplador. Velocidad de giro 200RPM (elaboración propia)</i>	<i>14</i>
<i>Figura 14 Oscilograma de las señales A y B con giro y encoder encendido a través del optoacoplador. Velocidad de giro 200RPM (elaboración propia)</i>	<i>14</i>
<i>Figura 15 Oscilograma de la señal de paso por cero a través del optoacoplador. Velocidad de giro 200RPM (elaboración propia)</i>	<i>15</i>
<i>Figura 16 Código de prueba de Arduino para la medición de posición (elaboración propia)</i>	<i>16</i>
<i>Figura 17 Salida por puerto serie a partir de la aplicación Arduino IDE (1000 pulsos por revolución) (elaboración propia)</i>	<i>16</i>
<i>Figura 18 Salida por puerto serie a partir de la aplicación Arduino IDE (500 pulsos por revolución) (elaboración propia)</i>	<i>17</i>
<i>Figura 19 Oscilograma de las señales A y B girando en sentido horario (elaboración propia)</i>	<i>18</i>
<i>Figura 20 Oscilograma de las señales A y B girando en sentido antihorario (elaboración propia)</i>	<i>18</i>
<i>Figura 21 Fragmento de código para la generación de pulsos de comprobación de posición (elaboración propia)</i>	<i>19</i>
<i>Figura 22 Fragmento de código para la medición de tiempo entre señales de paso por cero (elaboración propia)</i>	<i>19</i>
<i>Figura 23 Salida por puerto serie a partir de la aplicación Arduino IDE (elaboración propia)</i>	<i>20</i>
<i>Figura 24 Salida generada por la herramienta Serial Plotter de la aplicación Arduino IDE de posición de giro del eje en sentido horario (ensayo a 200RPM). (elaboración propia)</i>	<i>20</i>
<i>Figura 25 Salida generada por la herramienta Serial Plotter de la aplicación Arduino IDE de posición de giro del eje en sentido antihorario (ensayo a 200RPM). (elaboración propia)</i>	<i>21</i>

<i>Figura 26 Salida generada por la herramienta Serial Plotter de la aplicación Arduino IDE de posición de giro del eje y señal de posición general (ensayo a 200RPM). (elaboración propia)</i>	21
<i>Figura 27 Datos de potencia y par del motor de imanes permanentes del montaje (Yaskawa, <math>\Sigma</math>-II Series SGMH User Manual, 2003)</i>	22
<i>Figura 28 Oscilograma de las tensiones de fase y la señal de posición cuadrada (elaboración propia)</i>	23
<i>Figura 29 Visualización mediante matlab de datos exportados del osciloscopio a 200RPM (elaboración propia)</i>	24
<i>Figura 30 Visualización mediante Matlab de las formas de onda de las tensiones de fase filtradas para eliminar el ruido (elaboración propia)</i>	24
<i>Figura 31 Visualización de datos mediante Matlab del ensayo a 200RPM (elaboración propia)</i>	25
<i>Figura 32 Visualización de datos mediante Matlab del ensayo a 600RPM (elaboración propia)</i>	26
<i>Figura 33 Visualización de datos mediante Matlab del ensayo a 1000RPM (elaboración propia)</i>	26
<i>Figura 34 Comprobación del efecto de filtrado sobre la señal original (elaboración propia)</i>	27
<i>Figura 35 Análisis de frecuencia realizado en Matlab mediante la transformada de Fourier (elaboración propia)</i>	28
<i>Figura 36 Análisis de frecuencia de la señal filtrada mediante Matlab a través de la transformada de Fourier (elaboración propia)</i>	28
<i>Figura 37 Comparación de la señal filtrada mediante un filtro paso bajo implementado en Matlab con la señal original (elaboración propia)</i>	29
<i>Figura 38 Oscilograma de las señales A y B del encoder con el motor girando a 200 RPM tratadas con Matlab (elaboración propia)</i>	30
<i>Figura 39 Oscilograma de las señales A y B del encoder con el motor girando a 800 RPM tratadas con Matlab (elaboración propia)</i>	30
<i>Figura 40 Oscilograma de las señales A y B del encoder con el motor girando a 1000 RPM tratadas con Matlab (elaboración propia)</i>	31
<i>Figura 41 Oscilograma de la señal de paso por cero junto a la señal de posición generada por el microcontrolador (elaboración propia)</i>	33
<i>Figura 42 Oscilograma de las señales A (ch1), B (ch2) y de paso por cero (ch3) tomada en el osciloscopio (elaboración propia)</i>	33
<i>Figura 43 Captura de la salida por puerto serie de la aplicación Arduino IDE indicando los pulsos contados por revolución (elaboración propia)</i>	34
<i>Figura 44 Oscilograma de las tensiones de cada una de las fases R, S, T y de la señal de posición generada por el microcontrolador tratadas con Matlab (elaboración propia)</i>	34
<i>Figura 45 Esquema de las partes de un motor de imanes permanentes (Diagram)</i>	35
<i>Figura 46 Esquema de posición del eje con respecto a los diferentes polos (Diagram)</i>	36
<i>Figura 47 Secuencia de giro del eje con respecto a la forma de onda del campo magnético (Diagram)</i>	36
<i>Figura 48 Esquema vectorial (Lerma)</i>	37
<i>Figura 49 Esquema de ejemplo de motores con distintos números de polos (Diagram)</i>	38
<i>Figura 50 Esquema de control del motor simplificado (elaboración propia)</i>	39
<i>Figura 51 Esquema de control del motor simplificado con inversor (elaboración propia)</i>	40
<i>Figura 52 Esquema de control del motor simplificado con inversor y transformadas vectoriales (elaboración propia)</i>	40
<i>Figura 53 Esquema vectorial de corrientes en ejes alfa y beta (Lerma)</i>	41
<i>Figura 54 Esquema vectorial de corrientes en ejes d y q (Lerma)</i>	42
<i>Figura 55 Esquema de control del motor completo con referencias (elaboración propia)</i>	43
<i>Figura 56 Representación esquemática del circuito eléctrico de las fases del motor (elaboración propia)</i>	45
<i>Figura 57 Representación vectorial del flujo del eje del motor sobre los ejes d y q (elaboración propia)</i>	46
<i>Figura 58 Representación vectorial del flujo magnético del estator y del eje sobre los ejes d y q (elaboración propia)</i>	46

Figura 59 Modelado de las ecuaciones del motor en Simulink (corriente) (elaboración propia)	50
Figura 60 Modelado de las ecuaciones del motor en Simulink (corriente y par) (elaboración propia)	51
Figura 61 Implementación del control en Simulink con realimentación de corriente (elaboración propia)	52
Figura 62 Esquema de control del motor completo en Simulink (elaboración propia)	52
Figura 63 Implementación del control por corriente en Simulink con representación de datos (elaboración propia)	54
Figura 64 Captura de la simulación en Simulink (elaboración propia)	55
Figura 65 Esquema del control de velocidad implementado en Simulink (elaboración propia)	56
Figura 66 Captura de la simulación del control de velocidad en Simulink (elaboración propia)	57
Figura 67 Captura de la simulación del control de velocidad con segunda iteración de parámetros en Simulink (elaboración propia)	57
Figura 68 Captura de la simulación del control de velocidad con tercera iteración de parámetros en Simulink (elaboración propia)	58
Figura 69 Captura de la simulación de velocidad en Simulink con inciso en la sobreoscilación ante cambio en consigna (elaboración propia)	58
Figura 70 Captura de la simulación de velocidad en Simulink (segunda iteración de parámetros del controlador) con inciso en la respuesta ante cambios de referencia (elaboración propia)	59
Figura 71 Captura de la simulación de velocidad en Simulink (tercera iteración de parámetros del controlador) con inciso en la respuesta ante cambios de referencia (elaboración propia)	59
Figura 72 Captura de la simulación de velocidad en Simulink (cuarta iteración de parámetros del controlador) con inciso en la respuesta ante cambios de referencia (elaboración propia)	60
Figura 73 Captura de la simulación de velocidad en Simulink (quinta iteración de parámetros del controlador) con inciso en la respuesta ante cambios de referencia (elaboración propia)	60
Figura 74 Captura de la simulación de velocidad en Simulink con el controlador ajustado (elaboración propia)	61
Figura 75 Ejemplo de función de onda trifásica senoidal (elaboración propia)	61
Figura 76 Comparación entre $V_0$ y las ondas trifásicas senoidales (elaboración propia)	62
Figura 77 Suma de las formas de ondas trifásicas senoidales y $V_0$ (elaboración propia)	63
Figura 78 Secuencia de conmutaciones de las ramas del inversor (elaboración propia)	63
Figura 79 Esquema vectorial del vector de tensión en el primer sextante (elaboración propia)	64
Figura 80 Simulación en Matlab del control SVPWM (elaboración propia)	66
Figura 81 Captura de la función Serial Plotter de la aplicación Arduino IDE de los tiempos de activación de las distintas ramas del inversor (elaboración propia)	67
Figura 82 Registro de señales SVPWM generadas por el ESP32 sin filtrar (elaboración propia)	67
Figura 83 Registro de señales SVPWM generadas por el ESP32 filtradas (elaboración propia)	68
Figura 84 Foto de la tarjeta con los componentes soldados y numerados (elaboración propia)	69
Figura 85 Gráfica de linealidad tensión-Corriente (LEM)	71
Figura 86 Esquema de conexión del sensor de corriente (elaboración propia)	71
Figura 87 Caracterización del sensor LEM 2 (elaboración propia)	72
Figura 88 Caracterización del sensor LEM 1 (elaboración propia)	73
Figura 89 Gráfica de corriente leída por el sensor LEM sin filtro (elaboración propia)	74
Figura 90 Respuesta en frecuencia del filtro seleccionado (simulado con LT-Spice) (elaboración propia)	74
Figura 91 Gráfica de corriente leída por el sensor LEM (filtrada) (elaboración propia)	75
Figura 92 Calibración de corriente nula de los sensores de corriente (elaboración propia)	76
Figura 93 Lectura de corriente de los sensores LEM calibrados (elaboración propia)	76
Figura 94 Diagrama vectorial de los ejes d y q junto con su sentido de giro (elaboración propia)	77
Figura 95 Comparación de las tensiones de fase con la señal de posición (elaboración propia)	78
Figura 96 Respuesta de la corriente ante un cambio de referencia (elaboración propia)	82
Figura 97 Ajuste de exponencial atenuada a la respuesta real del sistema (elaboración propia)	83

<i>Figura 98 Respuesta del control de corriente ante un cambio en la referencia de tipo escalón (elaboración propia)</i>	84
<i>Figura 99 Respuesta del controlador ante un cambio de referencia en escalón con acción integral (elaboración propia)</i>	85
<i>Figura 100 Respuesta del controlador ante un cambio en escalón con acción integral actualizada (elaboración propia)</i>	85
<i>Figura 101 Respuesta del controlador ante un cambio en escalón con aumento de ganancia (elaboración propia)</i>	86
<i>Figura 102 Respuesta del controlador ante cambio en escalón con gran aumento en la ganancia (elaboración propia)</i>	86
<i>Figura 103 Respuesta final del controlador de corriente ante un cambio en escalón (elaboración propia)</i>	87
<i>Figura 104 Comparación de métodos de obtención de velocidad (elaboración propia)</i>	89
<i>Figura 105 Respuesta del control de velocidad con clara oscilación ante cambios en la referencia de tipo escalón (elaboración propia)</i>	90
<i>Figura 106 Respuesta mejorada del controlador sin acción integral (elaboración propia)</i>	91
<i>Figura 107 Respuesta mejorada del controlador con acción integral (elaboración propia)</i>	91
<i>Figura 108 Respuesta del controlador con aumento de la acción integral (elaboración propia)</i>	92
<i>Figura 109 Comparación de la respuesta del controlador de corriente ante un cambio en la consigna de velocidad (elaboración propia)</i>	93

## **ÍNDICE DE TABLAS**

<i>Tabla 1 Tiempos de conmutación de las ramas del inversor para el primer cuadrante</i>	65
<i>Tabla 2 Tabla de ensayos de desfase de ángulos</i>	78



# 1. INTRODUCCIÓN

El control de máquinas síncronas de imanes permanentes es vital para la utilización de las máquinas más comunes para tracción y movilidad en la industria, como brazos robot, servomotores industriales e incluso tracción industrial, naval y automovilística.

La tecnología utilizada en el proyecto gira en torno al microcontrolador, diseñado para IoT, “Internet de las cosas” por sus siglas en inglés, ESP32. Este microcontrolador incorpora la capacidad de comunicación vía Wifi y Bluetooth, así como características técnicas de gran potencia, como contar con 2 núcleos de trabajo independiente a 240MHz cada uno de ellos.

Esto permite que pueda ser utilizado para tareas de control rápidas, como el control de motores, a la vez que compagina, gracias a su segundo núcleo, las tareas de comunicación inalámbrica con el usuario, además de tener un precio muy competitivo y una gran comunidad a su alrededor que facilita enormemente el prototipado con el uso del mismo.

En el caso particular del presente proyecto se ha optado por una comunicación Bluetooth para el control del motor a través de aplicaciones genéricas de comandos vía puerto serie y se ha hecho énfasis en tratar de recoger la mayor precisión del montaje utilizado con este microcontrolador.

El microcontrolador en cuestión además consta de comunicación I2C, SPI y por puerto serie entre otras. Adicionalmente, el ESP32 es capaz de generar a través de múltiples salidas señales PWM de altas frecuencias y con suficiente resolución. También consta de múltiples entradas con capacidad de lecturas analógicas. Esto último será muy útil a la hora de la lectura de corriente, ya que los sensores utilizados son de efecto hall y transforman la corriente que los atraviesa en una señal de tensión que posteriormente trataremos de forma digital.

Como se ha comentado previamente, el microcontrolador ESP32 consta de una amplia comunidad a su alrededor que facilita esquemas de conexión y nociones básicas para la implementación del mismo en una tarjeta PCB (circuito impreso, por sus siglas en inglés “Printed Circuit Board”). Esto será de ayuda a la hora de diseñar la tarjeta final que albergará todos los componentes, ya que el diseño del circuito suplementario al microcontrolador, la placa de desarrollo ESP32 Devkit V1, está publicado en la página oficial del microcontrolador.

## 1.1. OBJETIVOS

El objetivo general de este trabajo es el diseño de un banco de ensayos de servos síncronos de imanes permanentes de técnica senoidal (Brushless AC, BLAC) en el que se utiliza como carga un accionamiento de la misma tecnología basado en componentes estándar que, además, proporcionará las señales a utilizar para la medida de la posición del eje del banco. Los componentes con los que se cuenta para el desarrollo de este banco de ensayos son el accionamiento previo, un motor idéntico al existente en el citado accionamiento y un inversor trifásico para prototipado. Por ello, el objetivo general se divide en tres objetivos principales:

- El diseño del sistema de realimentación de posición a partir de las señales disponibles en el accionamiento existente.
- El diseño del conjunto del sistema de control, incluyendo una tarjeta de control basada en microcontrolador con los sensores e interfaces adicionales necesarios para su integración con el servomotor a controlar y con el inversor para prototipado y el programa de control del microcontrolador que realice todas las funciones necesarias para dicho control: procesado de las señales de realimentación, realización de los bucles de control adecuados y las funciones de entrada y salida para recibir órdenes y enviar el estado del sistema al operador del banco.
- El diseño de una placa de circuito impreso adecuada para integrar todos los componentes incluidos en el sistema de control.

El promotor del proyecto, el grupo ISEE del Instituto Universitario de Ingeniería Energética, formado por profesores del Departamento de Ingeniería Eléctrica de la Universitat Politècnica de València, ha establecido además que este banco de debe diseñar para facilitar el aprendizaje de las técnicas de control de accionamientos BLAC, por lo que debe ser altamente configurable y debe permitir acceso a las magnitudes (entradas, salidas, parámetros) que definen el sistema de control.

## 1.2. ANTECEDENTES

Desde la aparición del primer motor eléctrico de corriente continua en 1832 por William Sturgeon y tras la aparición en el 1887 del motor de inducción de Nikola Tesla el uso de los motores eléctricos en la industria ha resultado vital. Los motores eléctricos resultan sencillos de utilizar, a la vez que constan de gran potencia y reducido tamaño. Estas características hacen que su utilización en cualquier tipo de industria, desde su concepción hasta el momento actual sea totalmente obligatoria.

Sin embargo, no es hasta la aparición de la electrónica de potencia que estos motores eléctricos perciben realmente una mejora en su funcionamiento. Con la llegada de la electrónica de potencia llegan de la mano los controladores de motor. Estos permiten la regulación en potencia y velocidad de los motores, haciéndolos aún más cómodos y útiles para la industria. A día de hoy los motores eléctricos se expanden desde la industria hasta la automoción, pasando por campos como las industrias ferroviarias y las navales.

La variación de velocidad y control de par, y en consecuencia potencia del motor eléctrico los vuelven candidatos idóneos para la tracción en distintas industrias. La más comúnmente conocida es la industria ferroviaria. La gran capacidad de carga de las locomotoras de mercancías

y la necesidad de control de las mismas, además de la capacidad de cargar generadores diésel a bordo les ha permitido implementar la tracción eléctrica durante un largo periodo de tiempo.

Los motores síncronos de imanes permanentes facilitan el trabajo del control de velocidad y posición, volviéndolos idóneos para su uso en industrias que requieran de posicionamiento preciso y velocidad estable. Además, estos motores constan de una gran densidad de potencia. Estos motores han sido normalmente utilizados en la industria en forma de servomotores industriales, normalmente para el movimiento de cabezales, grúas, poleas o brazos robot. En estas últimas décadas junto con la expansión del coche eléctrico estos motores han sido también utilizados en automoción para la tracción del vehículo. El inconveniente principal de estos motores siempre ha sido el alto coste de los materiales necesarios para la fabricación de los imanes permanentes (tierras raras) que residen en su rotor.

El control de altas prestaciones de este tipo de máquinas eléctricas hubiera sido imposible sin la aparición de la electrónica de potencia ni sin la rápida expansión de los microcontroladores, algunos especializados en el cálculo trigonométrico, que permiten un rápido control y ajuste de la corriente del motor y de su orientación respecto del rotor, permitiendo de esta manera el control de velocidad y posición tan preciso que requieren muchas de las aplicaciones donde este tipo de motores reinan.

Este trabajo toma como base un accionamiento industrial basado en una máquina BLAC y pretende compensar las dos principales limitaciones que este sistema industrial presenta desde el punto de vista del aprendizaje de accionamientos eléctricos: por una parte, la dificultad de acceso a las diferentes magnitudes que se utilizan en el sistema de control (por la limitación de entradas/salidas disponibles, de su velocidad de actualización y de las conexiones posibles con el sistema de control típicas de sistemas industriales) y, por otra, la rigidez del sistema de control, con una estructura definida por firmware en la que es posible introducir ciertos cambios muy limitados entre las conexiones de unos bloques constructivos con otras, pero que muestra siempre importantísimas limitaciones en relación al cambio de dicha estructura.

### **1.3. ESTRUCTURA DE LA MEMORIA**

Como se ha indicado, el primer paso para implementar el sistema de control de un servo BLAC es contar con la lectura de la posición angular del eje del motor, que se deberá obtener a partir de las señales proporcionadas por el accionamiento directamente acoplado a nuestro motor. Junto con esta medida podremos obtener una medida de velocidad del eje del motor. El diseño y comprobación paulatinos del sistema de realimentación de posición, por tratarse de un requerimiento previo para definir y diseñar el sistema de control y por su complejidad y extensión, se trata en el capítulo 2 de esta memoria.

El capítulo 3 de esta memoria recoge e toda su extensión el diseño del sistema de control, comenzando con un profundo estudio teórico del funcionamiento del motor y del diseño de los controladores para, una vez tener el funcionamiento claramente planteado, realizar un diseño preliminar del sistema mediante Simulink y Matlab incluyendo el análisis dinámico del motor y el diseño del criterio de control (orientación por campo del rotor) y de los lazos de control interno (corriente) y externo (velocidad) de la solución típica mediante lazos de control anidados en cascada. Una vez establecido y probado este diseño básico se abordará su diseño, implementación y pruebas en un sistema basado en microcontrolador, contemplando aspectos

como la operación del inversor (sistema de modulación de ancho de impulso, en nuestro caso modulación por vector espacial o SVPWM, dado que es el más extendido por los beneficios que aporta), la realimentación de corriente, y el diseño progresivo de los lazos de control de corriente y velocidad mediante reguladores PI discretos y su ajuste a las características del sistema a controlar.

El capítulo 4 de la memoria describe el diseño realizado de una placa de circuito impreso adecuada para incorporar todos los componentes necesarios del sistema de control diseñado y, por último, el capítulo 5 revisa los objetivos establecidos y los compara con los resultados contrastados alcanzados, constituyendo las conclusiones de este trabajo.

## 2. DISEÑO DE LA REALIMENTACIÓN DE POSICIÓN

### 2.1. Introducción

En este capítulo nos dedicaremos al diseño de la realimentación de posición del eje del motor de imanes permanentes que pretendemos controlar. Para ello haremos uso del encoder digital que contiene el servopack. Este encoder, como se explicará más adelante, genera una serie de pulsos a través de los cuales se puede deducir la posición en la que se encuentra el eje. Además de la interpretación de las señales también se realizarán pruebas a lo largo del capítulo a diferentes frecuencias de funcionamiento para garantizar la funcionalidad del sistema de realimentación de posición a la máxima velocidad de giro. Se llevarán a cabo pruebas con distintos sistemas de aislamiento de ruido y se comentarán durante el capítulo.

### 2.2. Descripción de las señales existentes

El promotor del proyecto ha especificado que la realimentación de posición debe utilizar las señales existentes en el controlador auxiliar (Yaskawa,  $\Sigma$ -II Series SGMH User Manual, 2003) que se va a utilizar para el movimiento del motor principal cuando éste no sea alimentado y para proporcionar un par resistente en el funcionamiento en carga del motor principal. De este Servopack extraeremos las señales que nos son indispensables para la realimentación de posición. Estas señales son las señales equivalentes a las que enviaría un encoder (codificador) óptico incremental en el eje del motor. El propio Servopack recibe en realidad las señales procedentes de un encoder absoluto de 13 bits codificadas por medio del puerto serie y genera y transmite las señales A, B y C(cero) típicas de un encoder óptico incremental por las salidas especificadas en el manual de usuario:

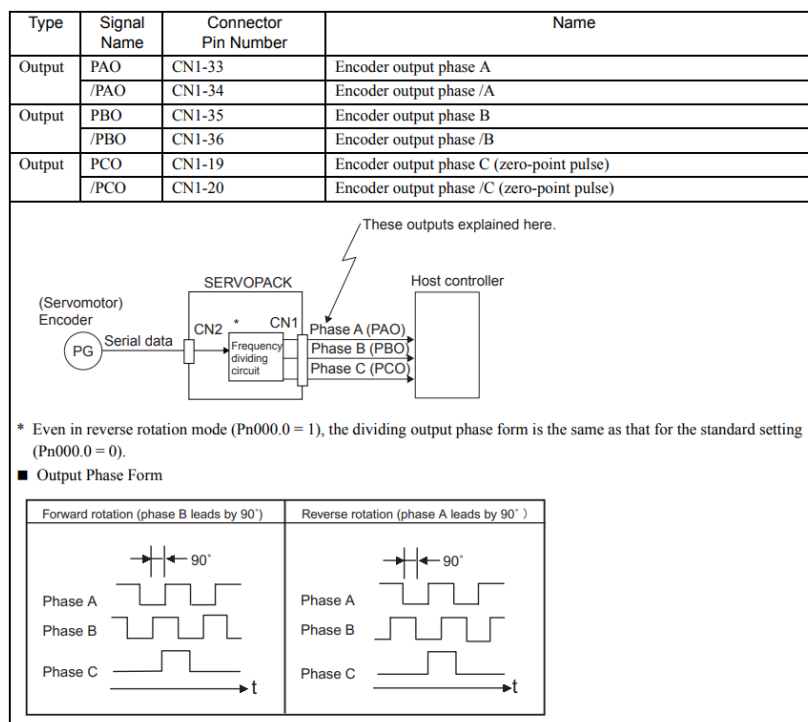
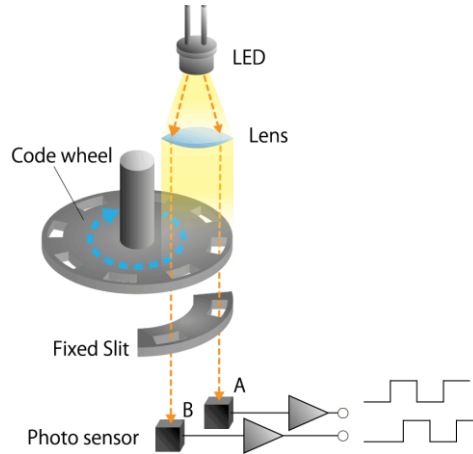


Figura 1 Descripción funcionamiento encoder interno (Yaskawa,  $\Sigma$ -II Series SGMH User Manual, 2003)

Como podemos observar las señales que nos interesan son las nombradas como PAO, /PAO, PBO, /PBO, PCO y /PCO. A las nombradas como PAO y su negada /PAO nos referiremos de ahora en adelante como A y /A y de manera similar con B y C. Las señales A y B nos serán útiles para extraer la posición del eje basándonos en su principio de funcionamiento (**¡Error! No se encuentra el origen de la referencia.**).



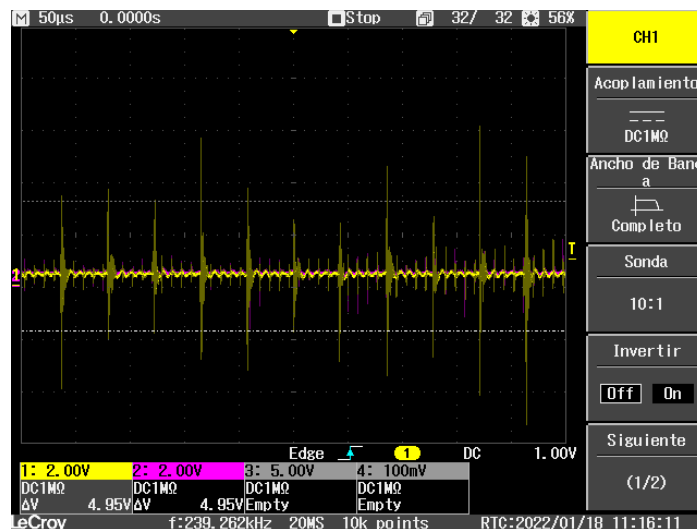
**Figura 2 Esquema de funcionamiento de un encoder óptico (AKM)**

Se explicará más adelante en la parte del código como extraeremos los pulsos por revolución a partir de este principio. Adicionalmente el Servopack también nos extrae mediante las señales C y /C el punto de paso por cero, es decir nos da una un pulso cada vez que el eje ha completado una vuelta completa.

En este capítulo revisaremos en detalle las señales proporcionadas por el Servopack y describiremos el proceso de diseño seguido para adaptar estas señales para su procesado por el sistema que va a efectuar las funciones de control, un ESP32.

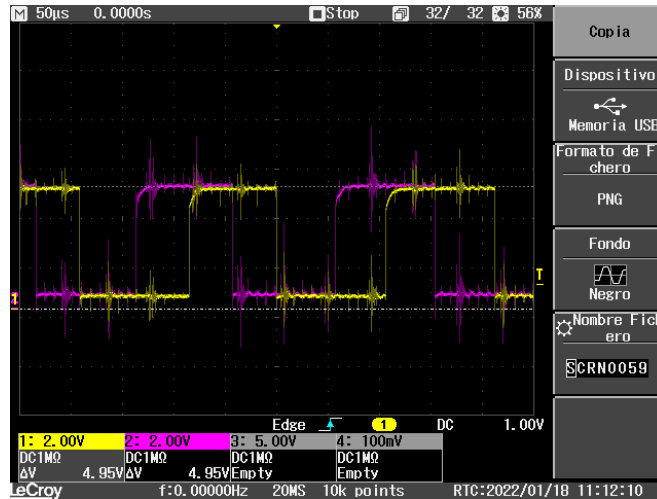
### 2.3. DESCRIPCIÓN DE SEÑALES PRODUCIDAS POR EL SERVOPACK

En este punto veremos el punto desde el que partimos y las señales originales que el Servopack envía.



**Figura 3 Oscilograma de la señal del encoder sin giro y servo encendido (elaboración propia)**

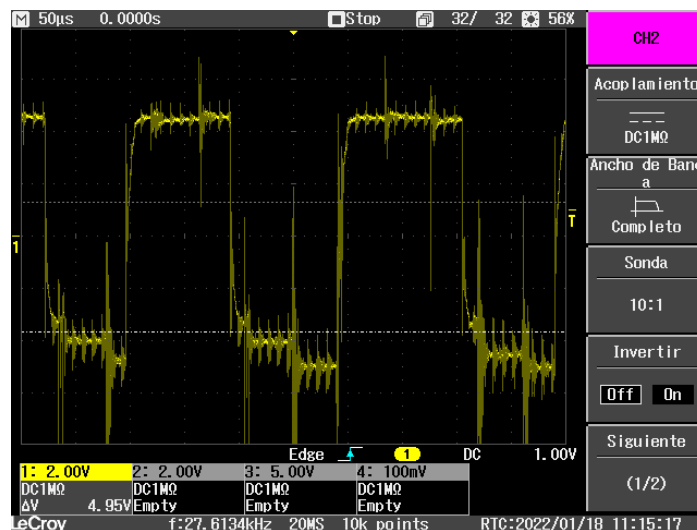
En esta primera imagen podemos observar cual es la señal que envía el Servopack para uno de los canales del encoder sin estar girando el motor. Se puede observar cómo, incluso parado el eje, la señal que nos llega es muy ruidosa con picos de tensión que sobrepasan los 5V. Este fenómeno observaremos a continuación que se suma a las señales A y B que queremos extraer del sistema.



**Figura 4 Oscilograma de la señal del enconder con giro y servo encendido (elaboración propia)**

En la figura 4 podemos observar las señales A y B, de forma decalada (los flancos de una señal coinciden con el centro de los periodos planos de la señal contraria) y con el efecto de ruido superpuesto. Además, podemos observar también que las señales, medidas entre PA0 y masa y entre PB0 y masa son señales de valor estable 5V. Debido a que nuestro objetivo es utilizar un ESP32 para hacer la lectura de las mismas y siendo este únicamente capaz de tolerar señales de hasta 3V en sus pines de entrada está claro que, como mínimo, necesitaremos de algún tipo de divisor de tensión para hacerlos compatibles.

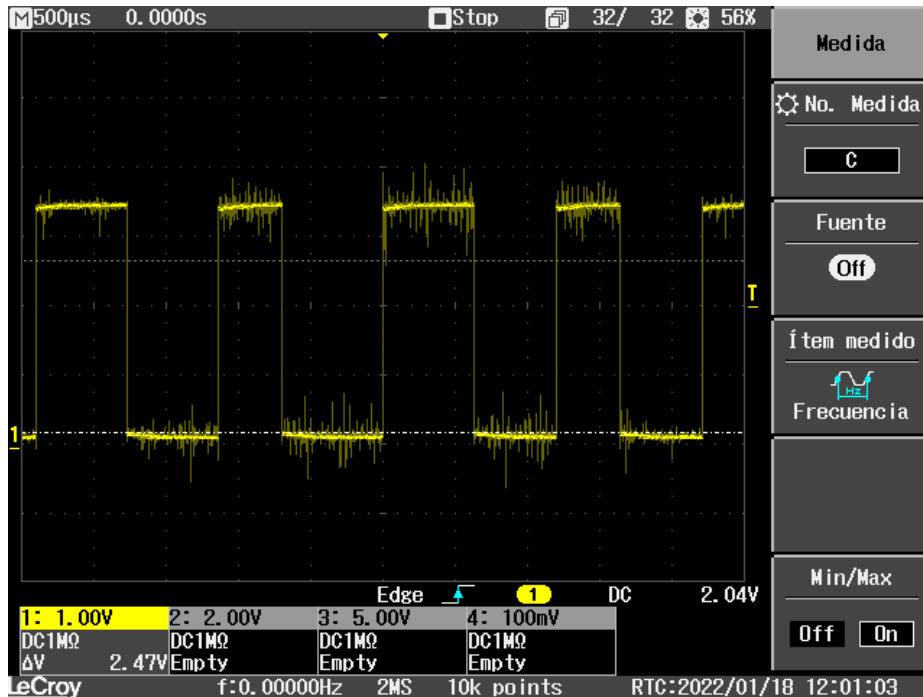
Adicionalmente también se hizo la medida de las señales A y /A conjuntamente para analizar si había algún tipo de reducción en el ruido base que encontrábamos al principio.



**Figura 5 Oscilograma de la señal A con giro y servo encendido con referencia a la señal /A (elaboración propia)**

Nuevamente observamos que en este caso nos encontramos ante una señal con valor medio nulo y con valores de 10V pico a pico con mucho ruido superpuesto.

Este ruido hay que remarcar que conocemos que proviene del funcionamiento activo del Servopack al hacer circular corriente mediante modulación PWM en las fases del motor, ya que, al extraer las señales con un movimiento manual del motor y el Servopack sin habilitar no observamos un ruido tan intenso.



**Figura 6 Oscilograma de la señal A con giro manual y referencia a masa (elaboración propia)**

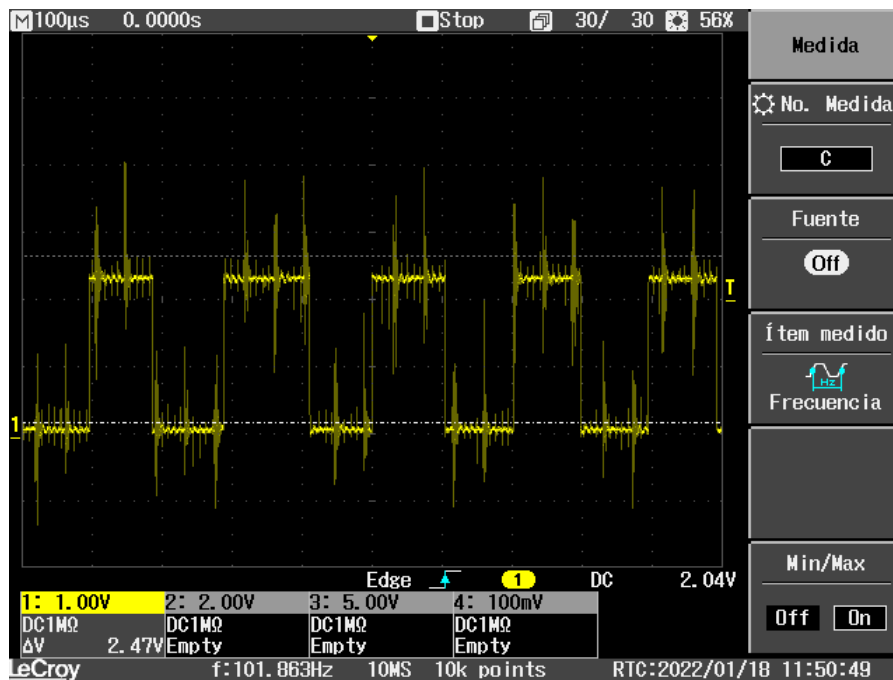
Como podemos observar en la figura 6, si giramos nosotros manualmente el eje del motor el ruido tan severo que observábamos antes desaparece.

En la figura 2 también observamos que los valores estables de tensión se encuentran a 5V, como ya se ha comentado y sin embargo moviendo manualmente el eje del motor observamos que estos valores pasan a estar cercanos a los 3V (figura 6).

De todas formas, puesto que nosotros queremos analizar el movimiento del motor por supuesto siendo accionado mediante el Servopack, en primera instancia tratamos de, mediante un divisor resistivo, reducir esta tensión estable hasta los 3V.



## 2.4. FILTRADO PASO BAJO Y DIVISOR RESISTIVO



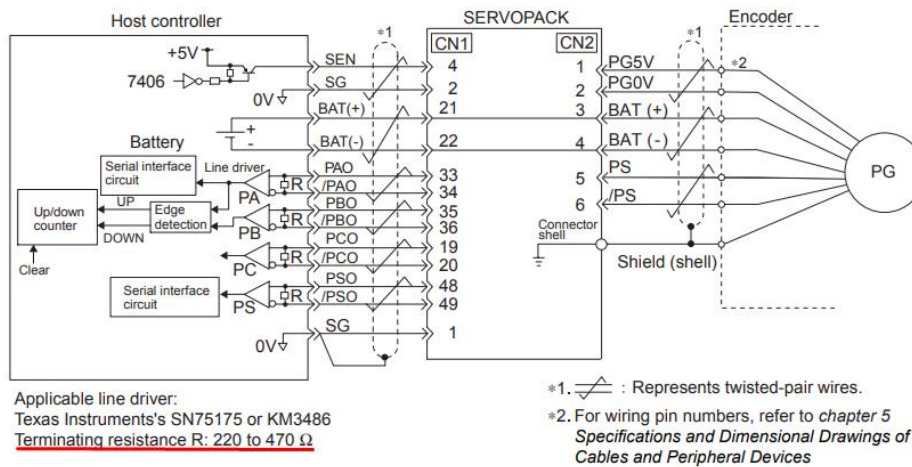
**Figura 7 Oscilograma de la señal A con giro y encoder encendido tras el filtrado (elaboración propia)**

En la figura 7 observamos como aun habiendo reducido el valor de la tensión en la parte estable hasta los 2.5V el ruido persiste. Esto nos lanzó a probar filtros del tipo paso bajo con combinaciones de condensador y resistencia para intentar reducir estos picos de tensión, pero las señales obtenidas fueron prácticamente iguales, únicamente perjudicando los tiempos de respuesta de los flancos de la misma.

Para tratar de corregir el ruido obtenido finalmente decidimos probar con un optoacoplador, de forma que separaremos las masas del Servopack y la del ESP32. Esto, además de ser necesario debido a los diferentes valores de la tensión de referencia que tenemos en el Servopack y en el ESP32, debería permitir eliminar perturbaciones de tensión, que son de poca energía y que no deben transmitirse a través del optoacoplador. Hay que señalar que el ruido presente en las figuras anteriores hace completamente inviable la utilización de esas señales con el ESP32 ya que producirán falsas detecciones de flanco en las señales del encoder y, por tanto, una incorrecta cuenta de pulsos.

## 2.5. OPTOACOPLADOR CON SALIDA TRANSISTOR

The following diagram shows the standard connections for an absolute encoder mounted to a servomotor. The connection cables and wiring pin numbers depend on the servomotor. For details, refer to *chapter 5 Specifications and Dimensional Drawings of Cables and Peripheral Devices*.



**Figura 8 Diagrama de conexiones de las salidas del encoder del servopack (Yaskawa, Σ-II Series SGMH User Manual, 2003)**

Como se ha comentado previamente la salida de las señales del Servopack es de  $\pm 5V$  con una carga típica mínima de  $220\Omega$ , lo que supone una corriente máxima de:

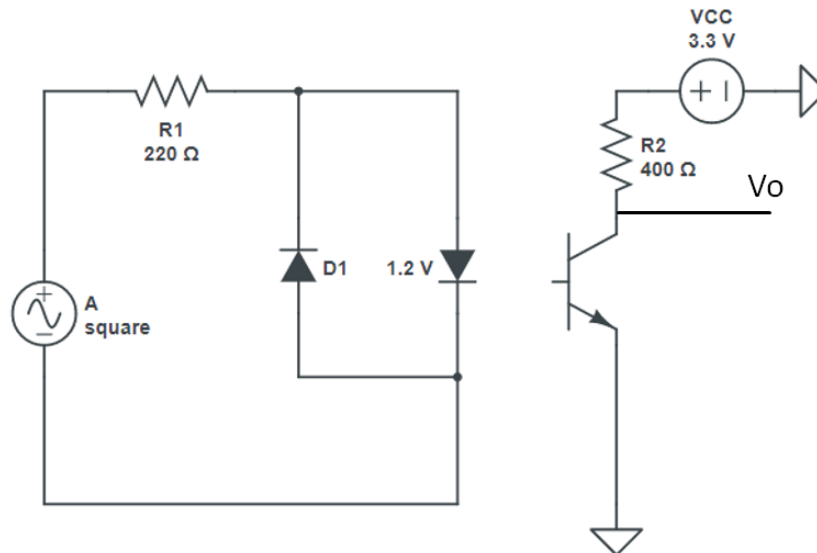
$$I_{PA} = \frac{5}{220} = 22mA$$

El optoacoplador seleccionado inicialmente a partir de estas especificaciones es el que tiene tiempos de transición más rápidos de entre los disponibles en el proveedor habitual (modelo ITV847 Liteon) aguanta hasta 6V en tensión inversa, por lo que sería conveniente el uso de un circuito de bloqueo de la parte negativa de la tensión, por ejemplo, un diodo en antiparalelo a la entrada.

La resistencia de línea de  $220\Omega$  está calculada para una tensión de  $+5V$ , como va a ir en serie con un diodo de unos  $1.2V$  Vf tenemos, para una  $I_F$  de  $20mA$ :

$$R = \frac{(5 - 1.2)V}{20mA} = 190\Omega$$

Si en lugar de  $190\Omega$  utilizamos una resistencia normalizada de  $220\Omega$  obtenemos una corriente  $I_F$  de  $17mA$ , aceptable según las especificaciones de entrada del optoacoplador.

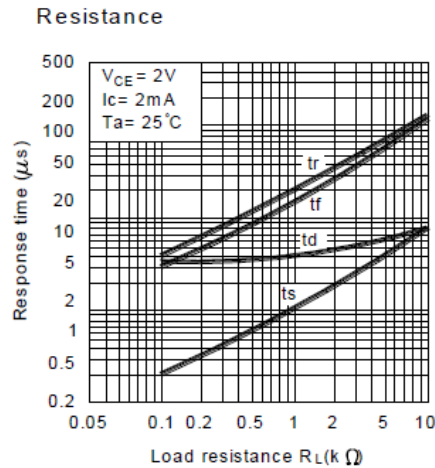


**Figura 9 Esquema de conexión del circuito de acoplamiento con optoacoplador con salida transistor (elaboración propia)**

Para tratar de minimizar el ruido anteriormente mencionado queremos hacer uso de la señal conjunta A-/A mediante un cable trenzado. Esta señal tal y como se ha mencionado previamente tiene valores pico a pico de 10V, es decir de  $\pm 5V$  con respecto a masa. Esto ocasionaría en el diodo del optoacoplador una tensión inversa de 10V. El optoacoplador no sería capaz de soportar esta tensión inversa, por lo que incluimos un diodo en antiparalelo de tensión directa inferior a los 6V máximos que podría soportar el optoacoplador. De esta forma estaremos limitando la tensión inversa del optoacoplador a la caída de tensión del diodo en antiparalelo. Lo mencionado anteriormente para la resistencia hace referencia a las pruebas realizadas en primera instancia, con la tensión simple de A con respecto a masa de 5V, por lo que la resistencia de 220Ω efectivamente limitará la corriente por debajo de los 20mA. En el caso de tener una señal con valores pico a pico de 10V la resistencia a utilizar sería casi el doble para poder limitar la corriente también por debajo de estos 20mA.

### 2.4.1. SELECCIÓN DEL NÚMERO DE PULSOS POR REVOLUCIÓN DEL ENCODER INTERNO AL SERVOPACK

Fig.10 Response Time vs. Load



Test Circuit for Response Time

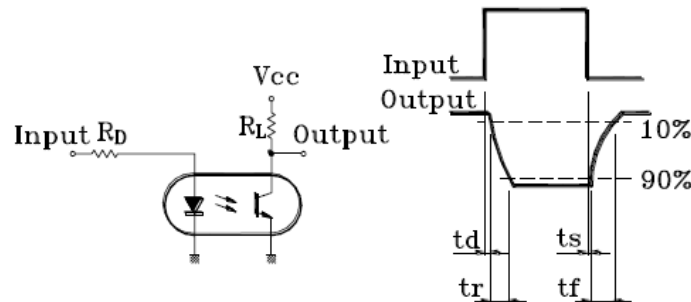


Figura 10 Esquema de conexión recomendada por el fabricante de optoacopladores (LiteON)

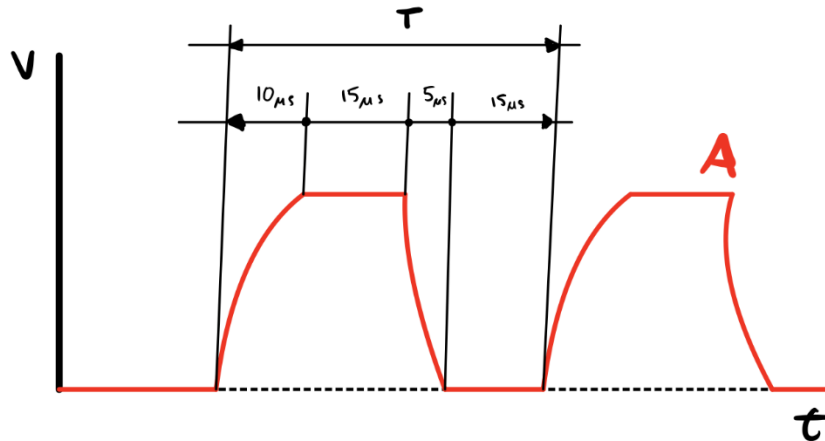
En este punto vamos a analizar las limitaciones en la frecuencia de los pulsos que surgen de los tiempos de transición positiva y negativa especificados para el optoacoplador.

Con una  $R_L$  de  $220\Omega$   $t_r$  y  $t_f$  salen de aproximadamente  $10\mu s$  y  $5\mu s$  respectivamente, lo que hace un total de unos  $15\mu s$  y de  $15\mu s$  para el pulso completo. Con un tiempo estable a nivel alto y bajo de  $15\mu s$  cada uno (por lo menos tan largo como el tiempo de subida-bajada), tenemos un tiempo mínimo de pulso de  $45\mu s$ , una frecuencia de pulsos no mayor de  $22.2kHz$ .

El servomotor tiene tres pares de polos ( $p=3$ ) que para  $50Hz$  supone  $1000rpm=16.7 rev/s$ , luego debemos limitar el número de pulsos por revolución del encoder a:

$$\frac{22222Hz}{16.7Hz} = 1330$$

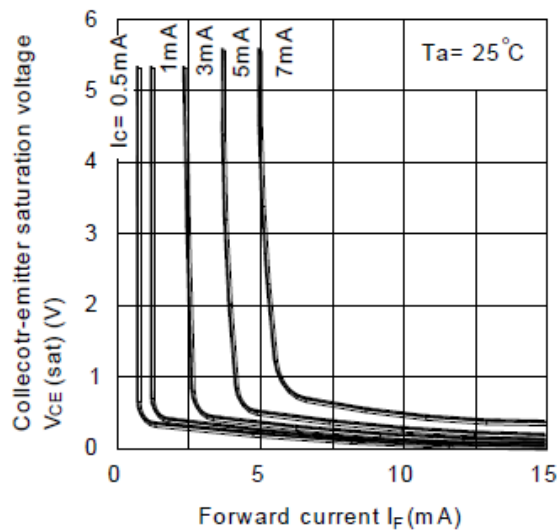
Un valor seguro sería de 1000 pulsos por revolución.



**Figura 11 Esquema de tiempos de la señal a través del optocoplador (elaboración propia)**

Para una  $I_F$  de unos 15-20mA tenemos una tensión de saturación del transistor de salida de menos de 0.5V:

**Fig.3 Collector-emitter Saturation Voltage vs. Forward Current**



**Figura 12 Gráfica de Voltaje-Corriente de la salida transistor del optocoplador (LiteON)**

Esto permitirá al ESP32 detectar esa caída como valor digital negativo. Dado que la alimentación del transistor la haremos desde la tensión de alimentación del ESP32 de 3.3V, debemos tratar de limitar la corriente de salida (para evitar problemas futuros al estresar el regulador de voltaje interno del módulo). Para limitar esa corriente a 7mA necesitaremos una resistencia pull-up de:

$$R_L = \frac{(3.3 - 0.5)V}{7mA} = 400\Omega$$

Siendo  $R_L$  la resistencia colocada tal y como se indica en el circuito de la figura 9.

### 2.4.2. SEÑALES CON OPTOACOPLADOR

En esta sección comprobamos la viabilidad del diseño establecido para el optoacoplador. Primero comencemos con las señales A y B. Para estos primeros ejemplos tomaremos la señal A.

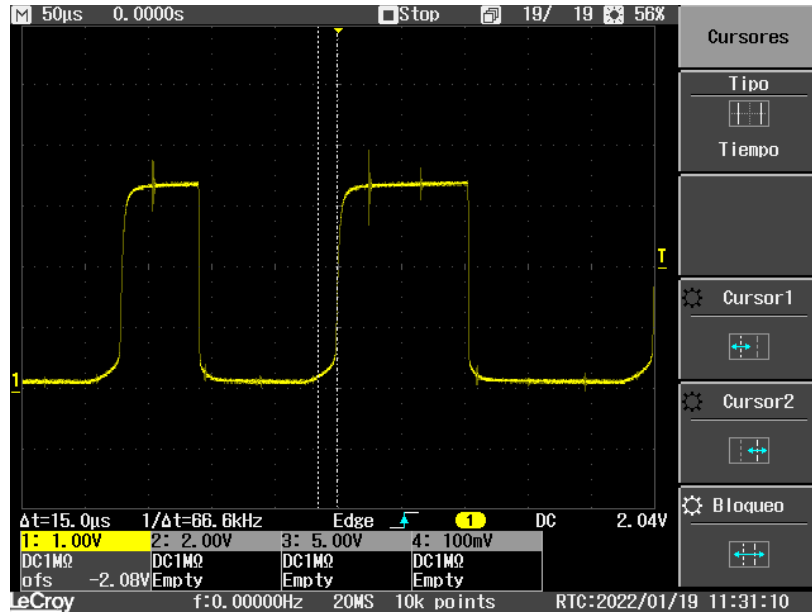


Figura 13 Oscilograma de la señal A con giro y encoder encendido a través del optoacoplador. Velocidad de giro 200RPM (elaboración propia)

En la figura 11 podemos apreciar como para la señal A, efectivamente con la introducción del optoacoplador hemos conseguido eliminar casi completamente el ruido que observábamos previamente. Además, se puede comprobar como la tensión pico a pico de nuestra señal es ahora de 3.3V, perfecta para el ESP32. En la figura 11 también podemos observar como el tiempo  $tr + td$  es de  $15\mu s$ . Veamos ahora como quedan las señales A y B solapadas:

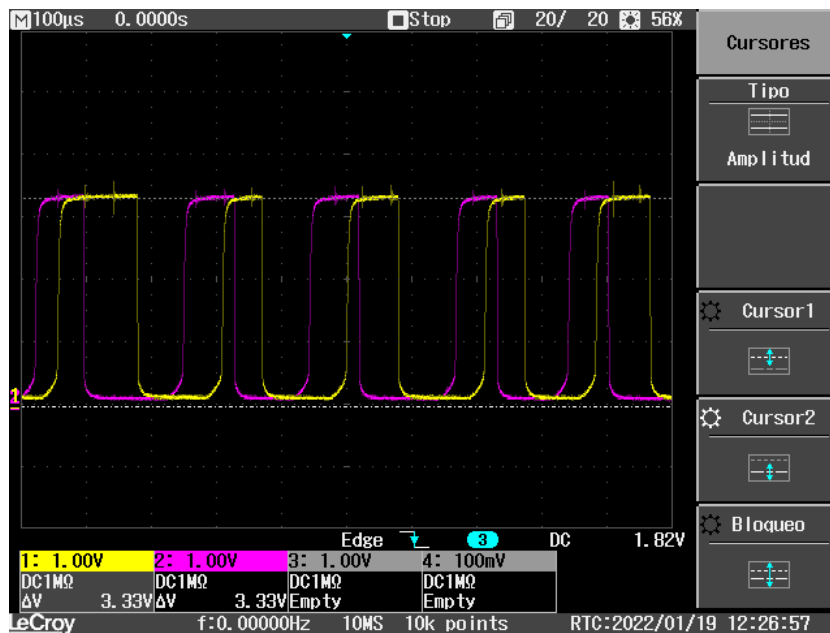
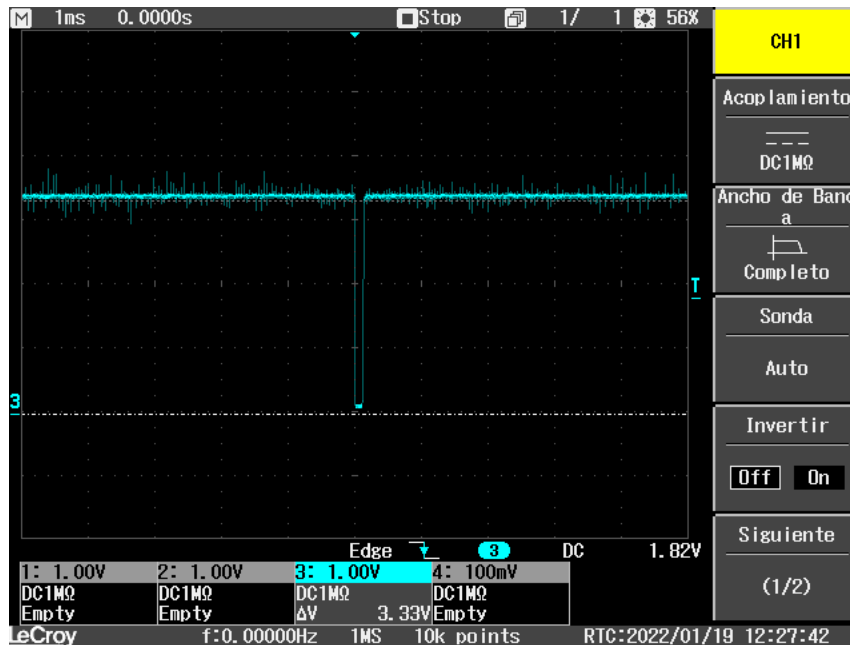


Figura 14 Oscilograma de las señales A y B con giro y encoder encendido a través del optoacoplador. Velocidad de giro 200RPM (elaboración propia)

Como podemos observar el ruido ha desaparecido prácticamente por completo. Además, podemos observar la similitud de las señales con la marcada en la figura 1. A partir de ahora podemos comenzar con el tratamiento de la señal vía el microcontrolador. Notar que en este caso y para facilitar el montaje, la señal ha sido obtenida del Servopack con referencia a masa. Esto quiere decir que el poco ruido encontrado en las figuras 14 y 13 aún podría ser reducido utilizando pares trenzados que son más inmunes al ruido radiado.

El optoacoplador utilizado cuádruple. Para nuestro proyecto utilizaremos 2 para las señales A y B y uno para la señal C, o de paso por 0.



**Figura 15 Oscilograma de la señal de paso por cero a través del optoacoplador. Velocidad de giro 200RPM (elaboración propia)**

### 2.4.3. CÓDIGO DE MEDIDA DE POSICIÓN Y PRIMERAS PRUEBAS

Una vez tenemos una señal limpia comenzaremos con el registro de datos vía ESP32. El circuito a utilizar es el mismo utilizado anteriormente para las medidas del osciloscopio. En este caso será el regulador de tensión del ESP32 el que proporcionará los 3.3V y GND para las medidas.

El primer paso será comprobar que, configurado el Servopack para mandar 1000 pulsos por revolución, nosotros leemos 1000 pulsos entre cada señal de (C). Para ello, asignaremos una interrupción a la señal de A cada vez que esta pase de HIGH a LOW (indicado como FALLING en el código) y reiniciaremos el contador cuando la señal C pase también de HIGH a LOW. El código queda así:

```

1 #define PINA 18 // Pin D18 del ESP32 será la entrada de la señal A vía optoacoplador
2 #define PINB 19 // Pin D19 del ESP32 será la entrada de la señal B vía optoacoplador
3 #define PINC 21 // Pin D21 del ESP32 será la entrada de la señal C vía optoacoplador
4
5 int veces = 0; // Señal que contará los pulsos de A
6
7 void IRAM_ATTR CUENTA() { // Función a ejecutarse durante el interrupt en A
8     veces=veces+1; // Cuenta los pulsos de A
9 }
10
11 void IRAM_ATTR EMPIEZA() { // Función a ejecutarse durante el interrupt en C
12     Serial.print("Veces_A= "); Serial.println(veces); // Muestra el contador en pantalla
13     veces=0; // Reinicia el contador
14 }
15
16 void setup() {
17     pinMode(PINA, INPUT); // Inicializamos los pines A y C como inputs
18     pinMode(PINC, INPUT);
19     Serial.begin(9600); // Comenzamos la comunicación serie a 9600 baudios
20     attachInterrupt(PINA, CUENTA, FALLING); // Inicializamos el interrupt en A y le atamos la función CUENTA
21     attachInterrupt(PINC, EMPIEZA, FALLING); // Inicializamos el interrupt en C y le atamos la función EMPIEZA
22 }
23
24 void loop() {
25
26 }

```

Figura 16 Código de prueba de Arduino para la medición de posición (elaboración propia)

El código nos muestra por el puerto serie estos datos:



```

COM3
Veces_A= 1018
Veces_A= 1017
Veces_A= 1009
Veces_A= 1013
Veces_A= 1013
Veces_A= 1008
Veces_A= 1013
Veces_A= 1006
Veces_A= 1009
Veces_A= 1009
Veces_A= 1015
Veces_A= 1008
Veces_A= 1016
Veces_A= 1008
Veces_A= 1013

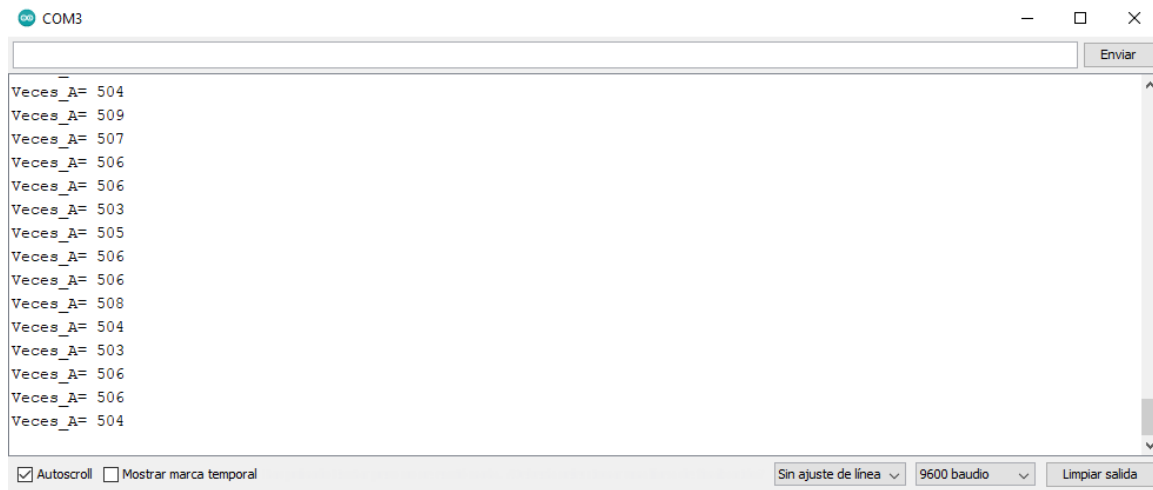
```

Figura 17 Salida por puerto serie a partir de la aplicación Arduino IDE (1000 pulsos por revolución) (elaboración propia)



Cómo podemos observar estamos contando alrededor de 1000 pulsos entre pulsación y pulsación de la señal C, el error en el conteo es relativamente pequeño (del 1%). Este error podrá ser ajustado más adelante.

Podemos comprobar también que, reduciendo en la configuración del Servopack los pulsos por revolución a 500 obtenemos estos datos:



**Figura 18 Salida por puerto serie a partir de la aplicación Arduino IDE (500 pulsos por revolución) (elaboración propia)**

Una vez más observamos que existe una relación estricta entre lo indicado por el servopack y lo que leemos a través del ESP32. Esto quiere decir que la corrección de ruido de la señal y la medida ha sido un éxito y que podemos continuar esta vez modificando el código para, con ayuda de la medida de la señal B, detectar también el sentido de giro.

#### 2.4.4. DETECCIÓN DEL SENTIDO DE GIRO Y ÁNGULO

Una vez comprobado que el conteo de pulsos por revolución funciona pasaremos al cálculo del ángulo de giro y sentido del mismo.

Para adivinar en qué sentido gira el eje nos tenemos que valer del principio de funcionamiento del sensor óptico, mostrado en las figuras 1 y 2. Para verlo con más claridad se han tomado capturas del osciloscopio girando en sentidos contrarios:



Figura 19 Oscilograma de las señales A y B girando en sentido horario (elaboración propia)

En este caso llamaremos a la señal 1 (marcada en amarillo) señal A, y a la señal 2 (marcada en rosa) señal B. Se puede observar que en la figura 18 cada vez que la señal A pasa de HIGH a LOW la señal B se encuentra en estado LOW. Esto ocurre para el sentido de giro nombrado arbitrariamente como 0 (sentido horario).

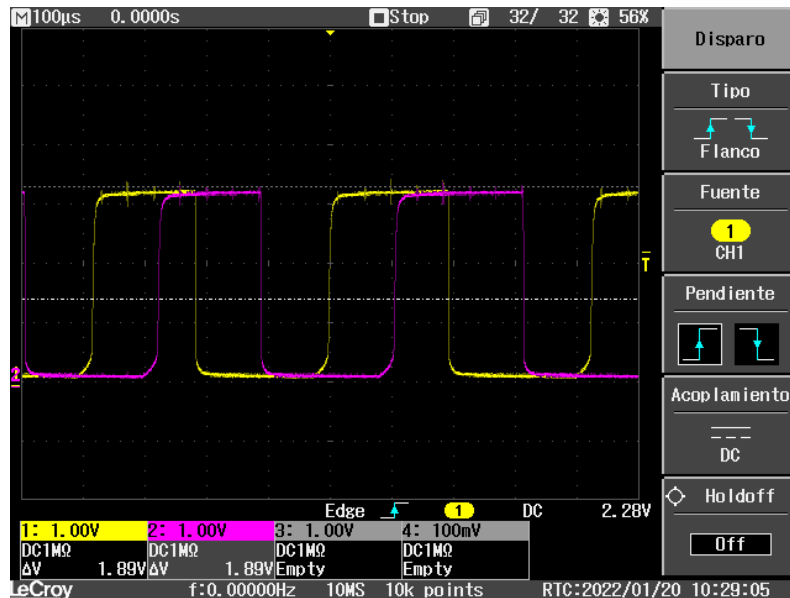


Figura 20 Oscilograma de las señales A y B girando en sentido antihorario (elaboración propia)

En la figura 19 se puede observar por el contrario que, al estar girando el eje en sentido contrario, esta vez cuando la señal A realiza ese cambio de estado de HIGH a LOW la señal B se encuentra ahora en estado HIGH. A este sentido le llamaremos pues, de forma también arbitraria, sentido 1 (sentido antihorario).

Visto esto está claro que, si cada vez que detectamos una pendiente negativa en la señal A la comparamos con la señal B podemos obtener dos posibles respuestas dependiendo del sentido de giro.

```
void IRAM_ATTR CUENTA() { // Función a ejecutarse durante el interrupt en A
  if(digitalRead(PINB){veces++;sentido=1;}
  else{veces--;sentido=0;}

  if(veces>=0 && veces<500) digitalWrite(PINSALIDA, HIGH);
  else digitalWrite(PINSALIDA, LOW);
}
```

**Figura 21 Fragmento de código para la generación de pulsos de comprobación de posición (elaboración propia)**

En el código de la figura 20 podemos ver qué es lo que ocurre cada vez que hay un cambio de HIGH a LOW en la señal A. Vemos que en cuanto eso ocurre el ESP32 lee directamente el estado lógico de la señal B. Si ocurre el estado de la figura 19 daremos el valor de 1 al sentido y aumentaremos el contador “veces”. En caso de que ocurra el estado de la figura 18 otorgaremos el valor 0 a la variable sentido y decrementaremos el contador de veces. Además, en esta función se lleva uno de los pines de salida del ESP32 a HIGH cada media vuelta y a LOW el resto de la vuelta, esto nos servirá más adelante para las medidas.

```
void IRAM_ATTR EMPIEZA() { // Función a ejecutarse durante el interrupt en C
  tiempo=micros()-tiempoprev; // Cuenta el incremento de tiempo entre revolución y revolución
  tiempoprev=micros();
  if(sentido) veces=0;
  else veces=1000;
}
```

**Figura 22 Fragmento de código para la medición de tiempo entre señales de paso por cero (elaboración propia)**

Veamos ahora la función que se ejecuta cada vez que se da un pulso de paso por 0 en la señal C. En esta función se calcula el tiempo entre revolución y revolución. Para explicar esto es necesario conocer cómo funciona la función micros(). Esta función cuenta el tiempo transcurrido en milisegundos desde que comienza la ejecución del código en el microcontrolador. Es una función que únicamente incrementa su valor. Lo que hacemos es guardar el valor de micros() en la última ejecución de la función y en la siguiente función restamos al valor actual de micros() el valor que habíamos guardado anteriormente. Esto nos dará el tiempo en milisegundos entre pulsos de paso por 0. Para calcular las revoluciones por segundo la fórmula sería:

$$rpm = \frac{1 \text{ rev}}{X\mu s} \cdot \frac{10^6 \mu s}{1 s} \cdot \frac{60 s}{1 \text{ min}}$$

Por último, una vez tenemos el conteo de “veces”, es decir de pulsos contados y conociendo el valor configurado en el servopack podemos calcular el ángulo girado. Para ejemplificarlo tomemos el valor configurado previamente en el servopack de 1000 pulsos por revolución:

$$\alpha = \frac{\text{veces}}{\text{pulsosxrev}} \cdot 365$$

Una vez hecho esto observemos los datos obtenidos por el puerto serie (puerto configurado a una velocidad de 115200, ya que a velocidades menores la precisión es muy buena):

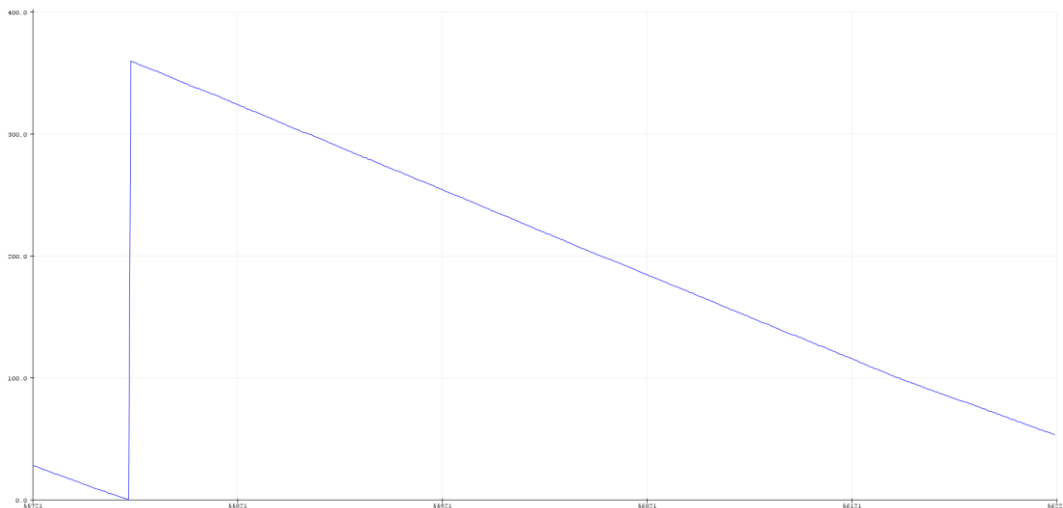
```

Angulo girado: 51.47  En sentido: 1
RPM = 150.38
Angulo girado: 55.48  En sentido: 1
RPM = 150.38
Angulo girado: 59.13  En sentido: 1
RPM = 150.38
Angulo girado: 63.14  En sentido: 1
RPM = 150.38
Angulo girado: 67.16  En sentido: 1
RPM = 150.38
    
```

**Figura 23 Salida por puerto serie a partir de la aplicación Arduino IDE (elaboración propia)**

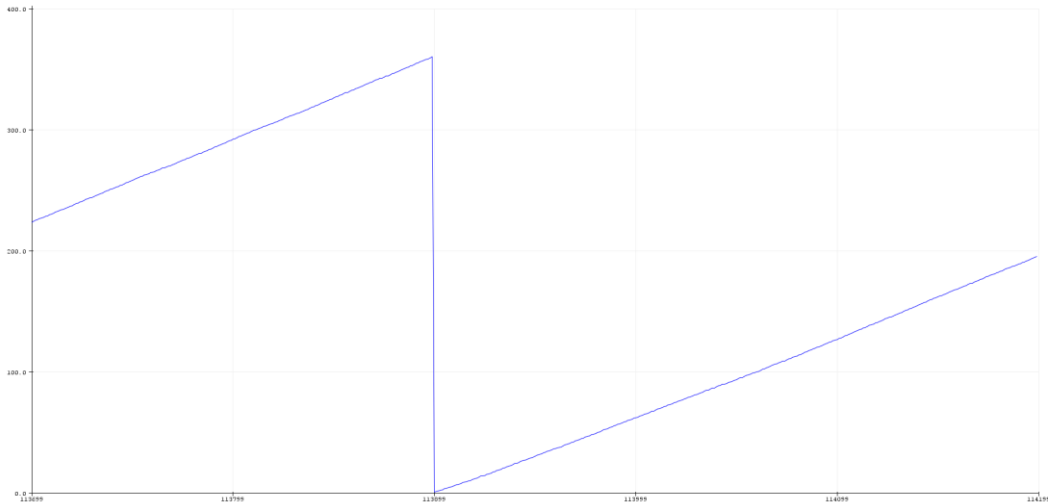
Podemos ver cómo, mientras que el motor gira, observamos que el ángulo crece (mostrando valores de ángulos no consecutivos debido a la falta de velocidad del puerto serie), que el sentido está claramente definido y que las revoluciones por minuto coinciden con bastante precisión con el valor marcado en el Servopack.

Además, en la función cada vez que el eje pasa por el punto de 0 dependiendo del sentido de giro se reinicia la variable veces a su valor máximo o mínimo.



**Figura 24 Salida generada por la herramienta Serial Plotter de la aplicación Arduino IDE de posición de giro del eje en sentido horario (ensayo a 200RPM). (elaboración propia)**

En la figura 24, obtenida haciendo uso de la herramienta “Serial Plotter” de la interfaz IDE de Arduino (motivo por el cual la fuente es de tan reducido tamaño y siempre haciendo el eje X referencia al tiempo interno del microcontrolador) observamos ahora de forma gráfica cómo actúa el valor de ángulo cada vez que pasa de 360º en la figura 23 cuando en ángulo decrece.

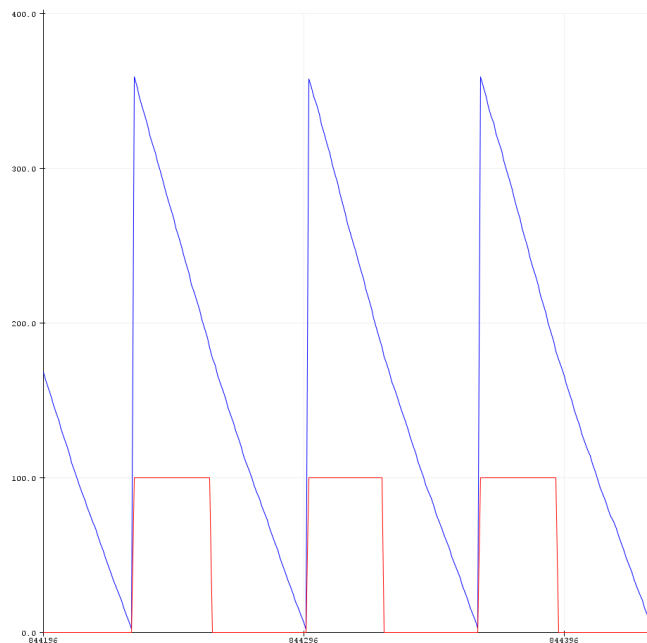


**Figura 25 Salida generada por la herramienta Serial Plotter de la aplicación Arduino IDE de posición de giro del eje en sentido antihorario (ensayo a 200RPM). (elaboración propia)**

Y en la figura 25 cómo actúa el valor cuando el ángulo crece.

A continuación, añadiremos una salida a uno de los pines del ESP32, esto nos servirá más adelante para tener una forma de comprobar en qué posición de giro estamos. Esta señal valdrá 1 durante media revolución y 0 durante la próxima mitad de la revolución.

Esto se visualiza en la figura 25. A continuación pasaremos a preparar el montaje para realizar los ensayos y medidas a distintas velocidades de rotación del motor, ensayos que utilizaremos para contrastar todo el sistema de adaptación y de procesamiento de las señales del encoder.



**Figura 26 Salida generada por la herramienta Serial Plotter de la aplicación Arduino IDE de posición de giro del eje y señal de posición general (ensayo a 200RPM). (elaboración propia)**

Observamos cómo durante la mitad de la rotación, entre 360 y 180 grados el ESP32 mandará una pulsación HIGH por uno de sus pines de salida. En la figura 26 esto se indica con una variable que, en salida HIGH vale 100 y en LOW vale 0.

### 2.4.5. ENSAYO DE TENSIONES DE FASE

Comenzaremos explicando el montaje y las medidas a realizar.

System Voltage	MOTOR SGMH-	Rated Output	Rated Torque*		Instantaneous Peak Torque*		Continuous Rated Current	Maximum Peak Current	Rated Angular Acceleration	Rated Power Rate*
		W (hp)	oz · in	N · m	oz · in	N · m	$A_{rms}$	$A_{rms}$	rad/s <sup>2</sup>	kW/s
100V ac	A3B	30 (0.04)	13.5	0.095	40.5	0.29	0.66	2.0	57500	3.49
	A5B	50 (0.07)	22.6	0.159	67.7	0.48	0.95	2.9	72300	11.5
	01B	100 (0.13)	45.1	0.318	135	0.96	2.4	7.2	87400	27.8
	02B	200 (0.27)	90.1	0.637	270	1.91	3.0	9.0	60100	38.2
	04A**	400 (0.53)	181	1.27	542	3.82	2.8	8.5	73600	93.7
200V ac	A3A	30 (0.04)	13.5	0.095	40.5	0.29	0.44	1.3	57500	5.49
	A5A	50 (0.07)	22.6	0.159	67.7	0.48	0.64	2.0	72300	11.5
	01A	100 (0.13)	45.1	0.318	135	0.96	0.91	2.8	87400	27.8
	02A	200 (0.27)	90.1	0.637	270	1.91	2.1	6.5	60100	38.2
	04A	400 (0.53)	181	1.27	542	3.82	2.8	8.5	73600	93.7
	08A	750 (1.01)	338	2.39	1010	7.1	4.4	13.4	35600	85.1

Figura 27 Datos de potencia y par del motor de imanes permanentes del montaje (Yaskawa, Σ-II Series SGMH User Manual, 2003)

Nuestro servomotor es el modelo SGMH-02A, necesitamos los datos de par nominal y corriente nominal. En nuestro caso el par nominal es  $T_n = 0.637Nm$  y la corriente nominal  $I_{ef} = 2.1A$ .

La constante de par  $kT$  se calcula como:

$$kT = \frac{T_n}{I_{est}} = \frac{0.637Nm}{2.1 \cdot \sqrt{2} A} = 0.214 \frac{Nm}{A_{fasor}}$$

La constante de par es par/amperio de fasor espacial, que es  $I \cdot \sqrt{2}$ , ya que I es la corriente eficaz.

Por teoría, la constante de fem es  $\frac{2}{3}$  de la constante de par.

$$K_E = \frac{2}{3} \cdot kT = \frac{2}{3} \cdot 0.214 = 0.143 \frac{U_{max}}{rad/s}$$

A 3000 rpm, que son  $100\pi \frac{rad}{s}$  tenemos una  $U_{max} = 45V$ , es decir una  $U_{ef} = 31V$ . Un ensayo realizado en 2012 por el departamento muestra un valor coherente: para tensión de línea en vacío se ven unos 80V a 3000rpm, siendo  $80V \cong 45 \cdot \sqrt{3}$  (para pasar a tensión de línea).

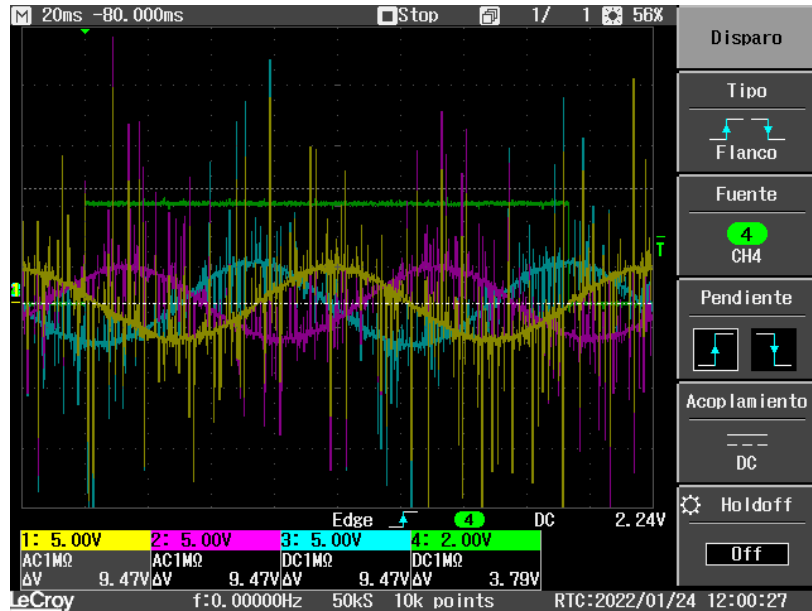
Para medir directamente tensiones de fase neutro, que son más fáciles de correlacionar en cuanto a su desfase con la posición del rotor que las de línea, vamos a conectar los terminales de salida del servomotor a una carga resistiva en estrella de alta impedancia, de forma que las tensiones fase-centro de la estrella coincidirán con las tensiones fase neutro (no accesible) del motor. A la velocidad máxima de 3000 rpm, con resistencias de 0.25W podríamos poner en estrella 3 resistencias:

$$R_{estrella} = \frac{U_{ef}^2}{P_{res}} = \frac{31^2 V^2}{0.25 W} = 3.8k\Omega$$

Luego con tres resistencias de 4k7 ohmios podemos realizar el ensayo.

En esas condiciones la corriente consumida es de  $\frac{31V}{4700\Omega} = 6.6mA$ , alrededor del 0.3% de la corriente nominal del servo, suficientemente baja como para que la caída de tensión en las fases (interna del servo) sea despreciable respecto de la fem inducida.

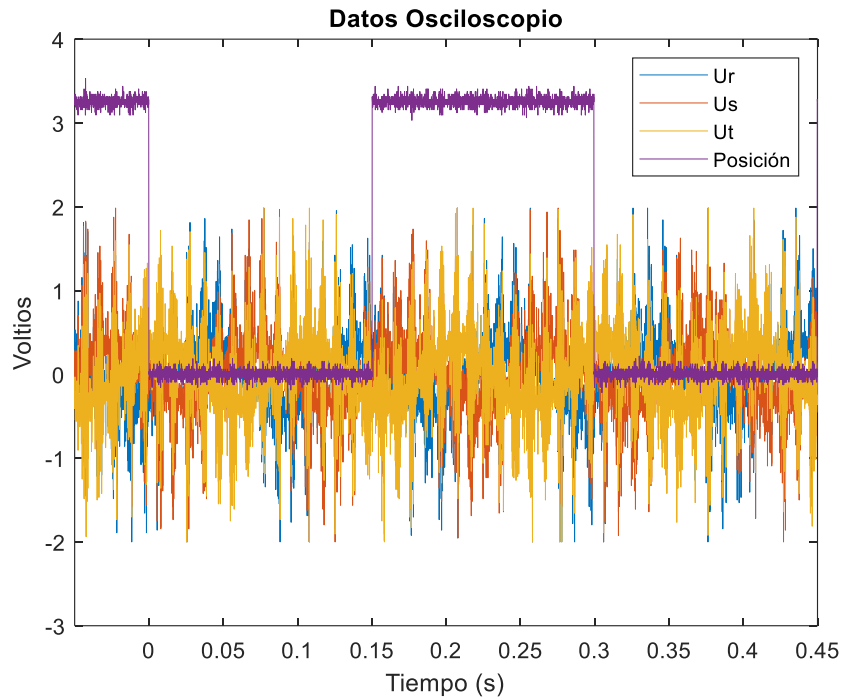
Realizamos el montaje indicado y tomamos las medidas pertinentes con el osciloscopio.



**Figura 28 Oscilograma de las tensiones de fase y la señal de posición cuadrada (elaboración propia)**

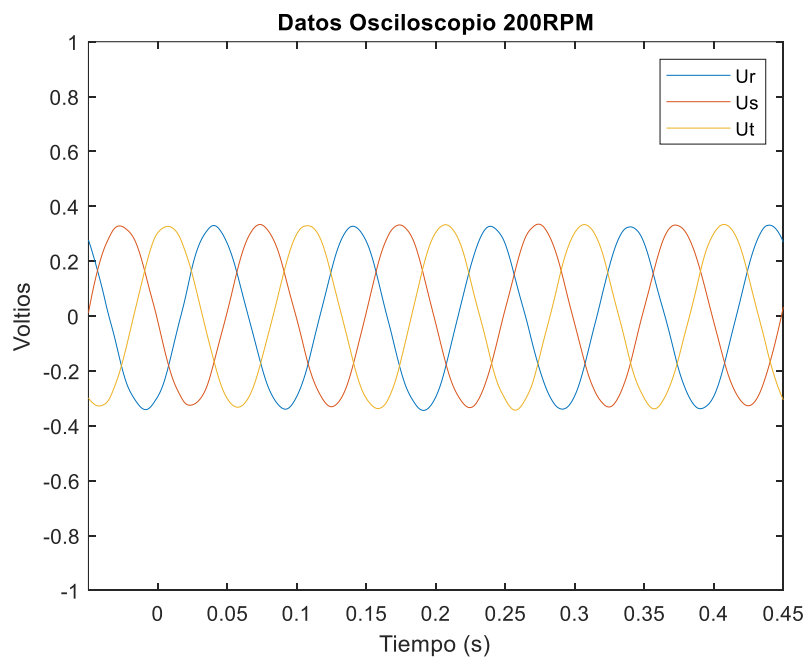
En este caso la medida de la figura 28 es solo un ejemplo de cómo se ven las señales medidas en el osciloscopio y nos servirá para ilustrar a qué hace referencia cada canal. Posteriormente el análisis más en detalle se hará mediante Matlab.

En este caso las señales en los canales 1,2 y 3 son las tensiones de fase con respecto al neutro de la estrella realizada con las resistencias. El canal 4 muestra el pulso que creamos en el código para mostrar la duración de una rotación del motor.



**Figura 29 Visualización mediante matlab de datos exportados del osciloscopio a 200RPM (elaboración propia)**

En la figura 29 podemos ver los datos obtenidos mediante el fichero .WFM del ensayo a 200 rpm. Este fichero lo podemos analizar mediante una función proporcionada por el departamento en Matlab. De esta forma es mucho más sencillo analizar los datos. El problema es, como se puede ver a simple vista el ruido. Para esto filtraremos la señal, que finalmente quedará así:



**Figura 30 Visualización mediante Matlab de las formas de onda de las tensiones de fase filtradas para eliminar el ruido (elaboración propia)**



Se puede ver ahora de forma mucho más clara en la figura 30 los valores de las tensiones. En este caso las tensiones están reducidas en un factor de 10, ya que se tomaron los datos con una sonda con configuración x10. En los siguientes ensayos mostrados se ajustan las tensiones para mostrar el valor real.

Mostraremos ahora algunos de los ensayos realizados a distintas velocidades:

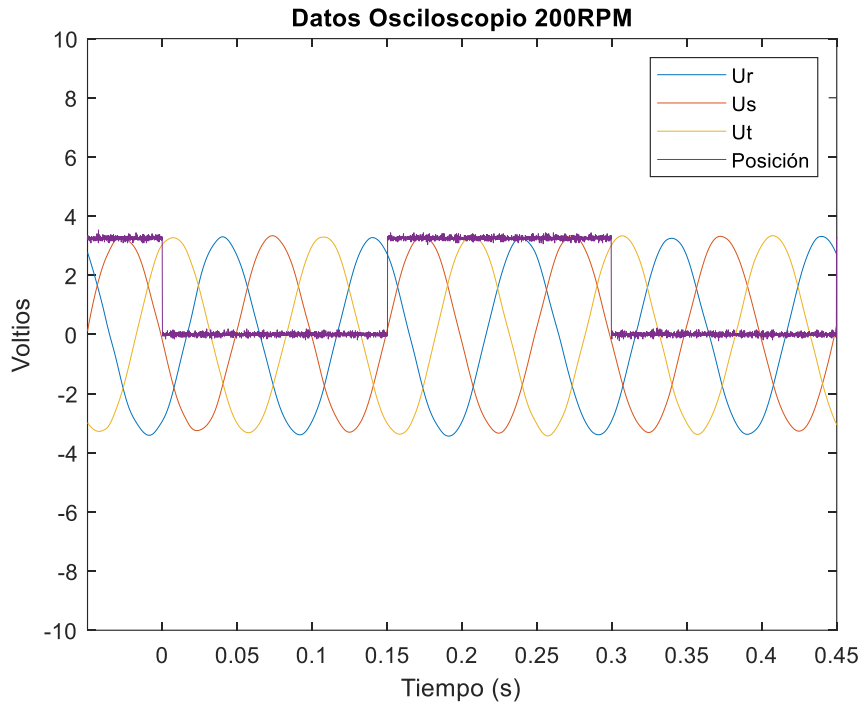


Figura 31 Visualización de datos mediante Matlab del ensayo a 200RPM (elaboración propia)

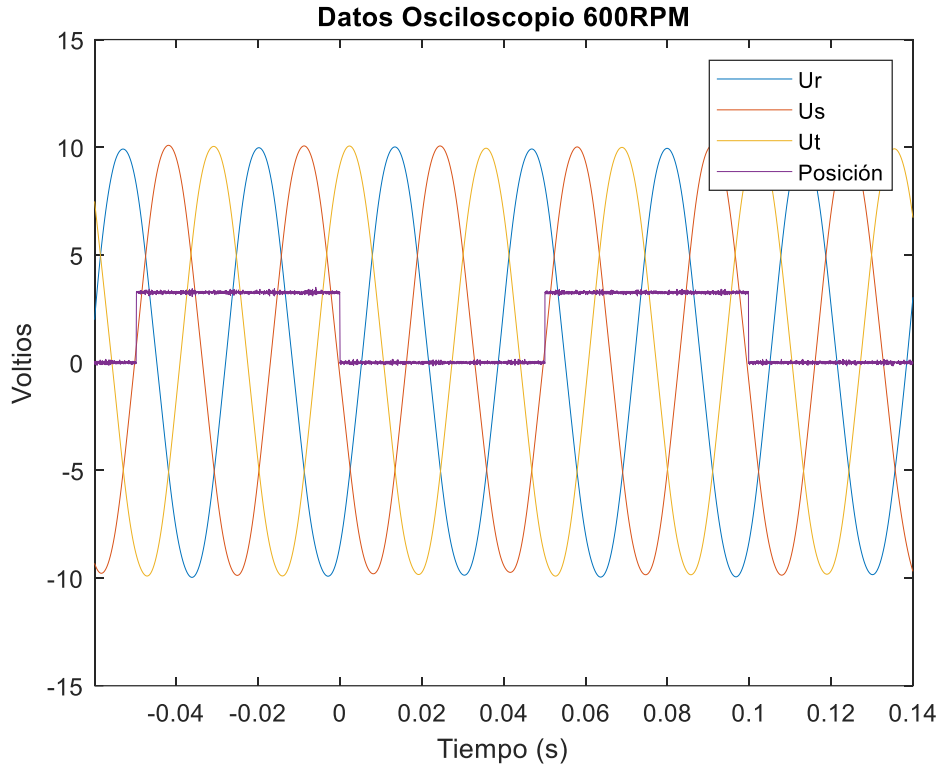


Figura 32 Visualización de datos mediante Matlab del ensayo a 600RPM (elaboración propia)

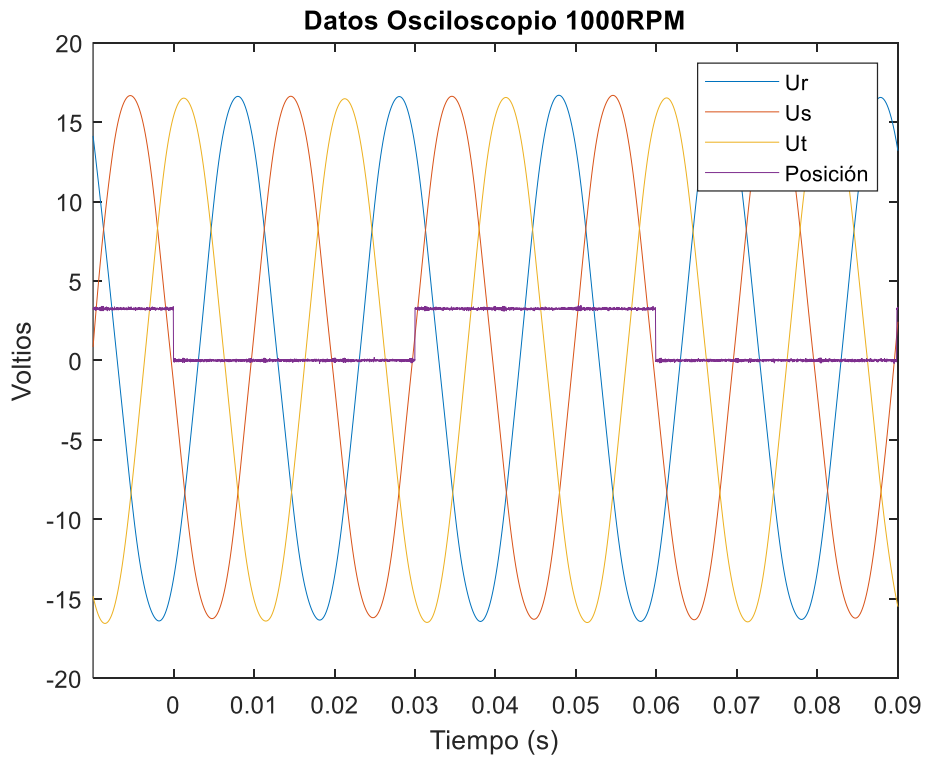
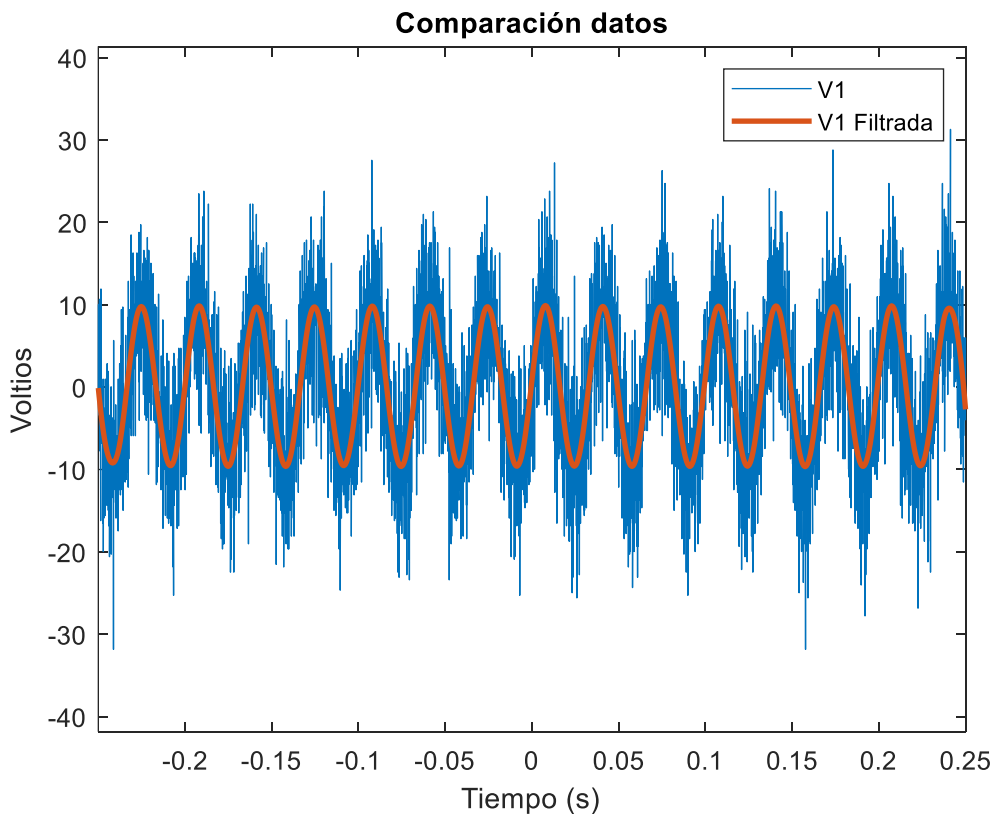


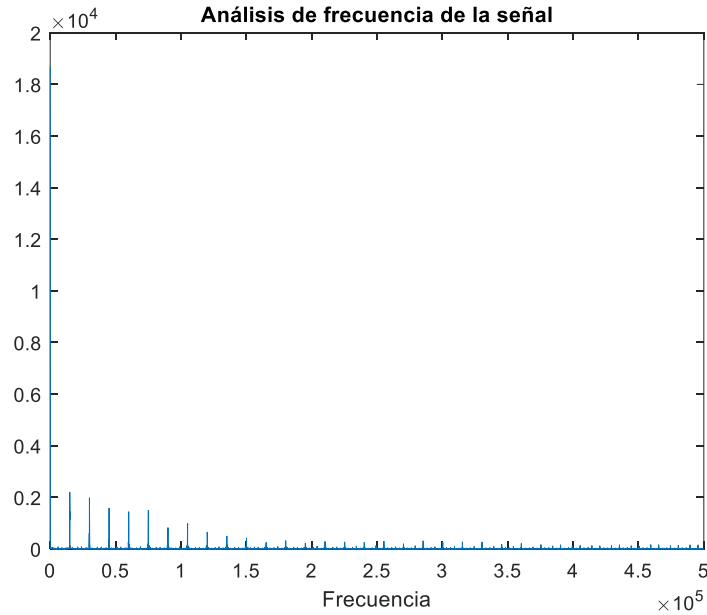
Figura 33 Visualización de datos mediante Matlab del ensayo a 1000RPM (elaboración propia)

La figura 35 muestra de forma patente un problema que será necesario investigar y resolver: aunque las señales senoidales de f.e.m. inducida presentan una gran uniformidad y estabilidad a todas las velocidades ensayadas, en esta figura se aprecia claramente una diferencia significativa entre la duración de la parte baja de la onda cuadrada de posición y la de la parte alta. Esto no parece muy razonable, ya que parecería indicar que la velocidad de giro de la máquina sería más alta durante cada segunda mitad de una revolución que durante la primera mitad. Esa diferencia no es coherente con la observación realizada de una f.e.m. de amplitud muy estable en todos sus periodos y en todas las fases. Por ello debemos pensar en un problema en el diseño del procesado (electrónico o por software) de las señales del encoder, problema que abordaremos en el siguiente epígrafe.



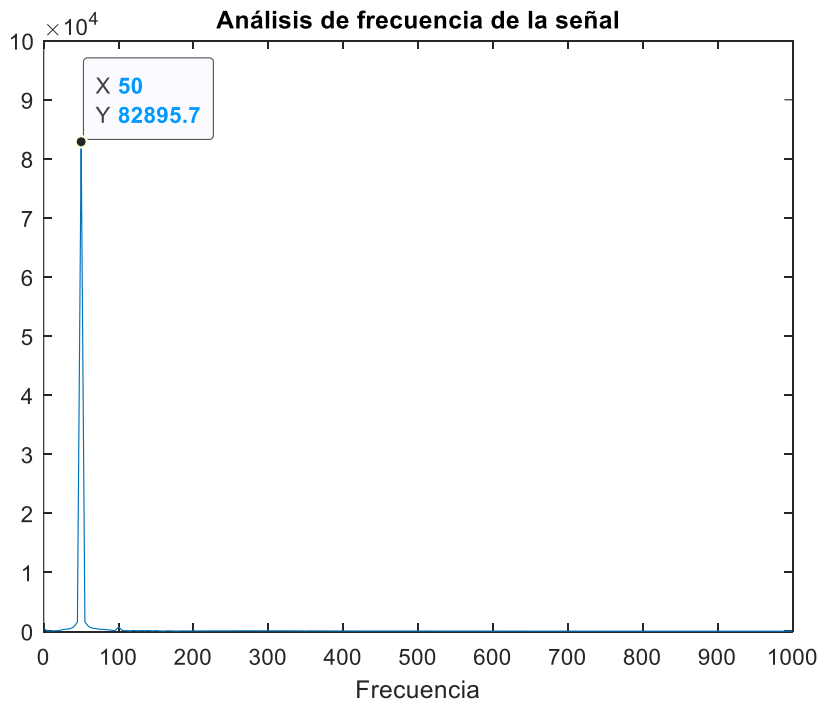
**Figura 34 Comprobación del efecto de filtrado sobre la señal original (elaboración propia)**

Adicionalmente en la figura 36 se muestra una pequeña muestra de cómo los valores obtenidos mediante las funciones de filtrado de ruido son de muy buena calidad y por tanto sus valores muy fiables.



**Figura 35 Análisis de frecuencia realizado en Matlab mediante la transformada de Fourier (elaboración propia)**

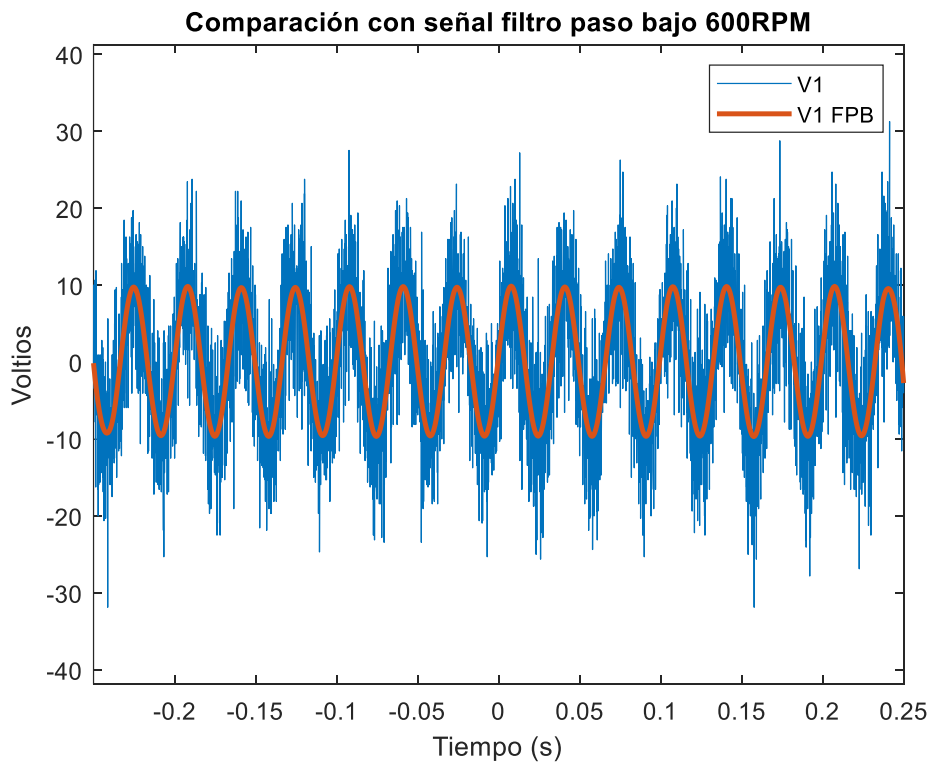
Podemos realizar un análisis espectral de la señal mediante la transformada de Fourier para obtener datos sobre las frecuencias predominantes de la señal y poder filtrarlas. Observamos que el análisis de frecuencia por Fourier nos muestra que tenemos fuentes de ruido hasta frecuencias del orden de los kHz, cuando realmente, nuestra frecuencia fundamental está en los 50 Hz. Si aplicamos el filtro anteriormente mencionado y repetimos el análisis de Fourier observamos un cambio drástico:



**Figura 36 Análisis de frecuencia de la señal filtrada mediante Matlab a través de la transformada de Fourier (elaboración propia)**

Como observamos y como cabía esperar, la frecuencia fundamental de la señal es de 50Hz.

Para implementar en el ESP32 un filtro, debemos optar por alguno más sencillo. Una buena opción podría ser un filtro paso bajo, que se muestra en la figura 39.



**Figura 37 Comparación de la señal filtrada mediante un filtro paso bajo implementado en Matlab con la señal original (elaboración propia)**

En la figura 39 observamos la comparación de la señal inicial con la señal una vez aplicado el filtro paso bajo, en este caso con frecuencia de corte de 50Hz. El filtro se ha aplicado dentro del propio código de Matlab, pero sirve como prueba de que un filtro de esas características puede ser útil para eliminar el ruido de la señal en el futuro.

## 2.6. REDISEÑO DE LOS OPTOACOPADORES

Para analizar el origen de la diferencia de duraciones entre la parte baja y alta de la señal de posición comentada en el epígrafe anterior, se registraron las señales a la salida de los optoacopladores en ensayos a velocidad constante y varias velocidades. Mientras que en la figura 40 se observa la secuencia típica de las señales AB del encoder, con flancos de cada una centrados más o menos en la zona plana de la señal contraria, las medidas a velocidades mayores muestran un problema en la transmisión de la señal por los optoacopladores que hace que la parte alta sea más breve que la baja.

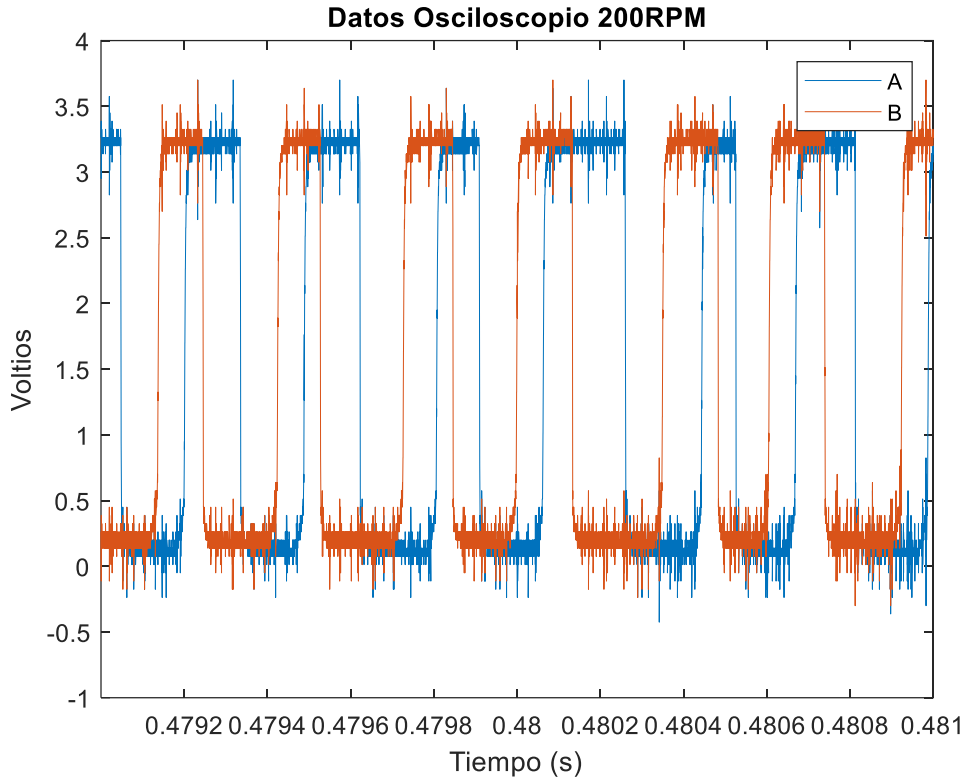


Figura 38 Oscilograma de las señales A y B del encoder con el motor girando a 200 RPM tratadas con Matlab (elaboración propia)

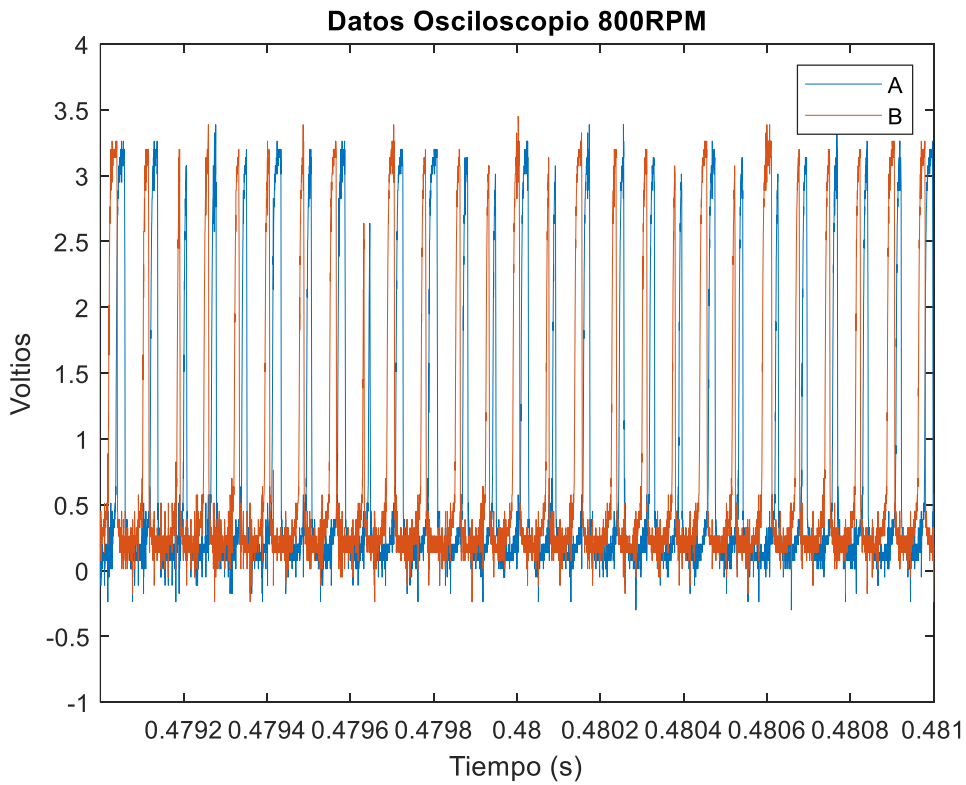
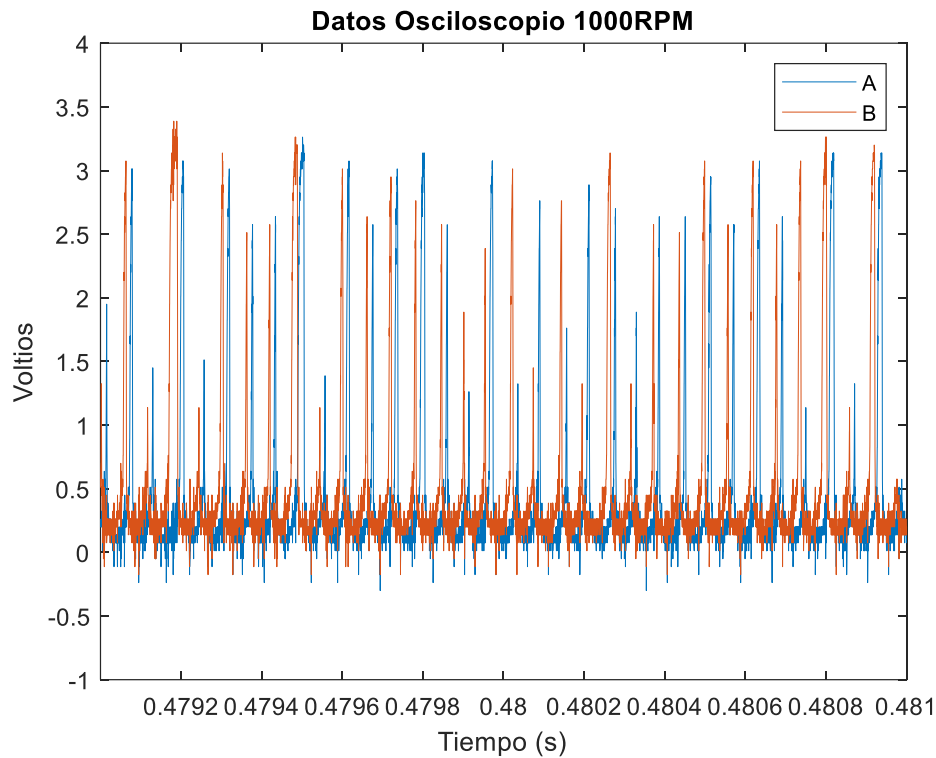


Figura 39 Oscilograma de las señales A y B del encoder con el motor girando a 800 RPM tratadas con Matlab (elaboración propia)



**Figura 40 Oscilograma de las señales A y B del encoder con el motor girando a 1000 RPM tratadas con Matlab (elaboración propia)**

Comentemos los problemas encontrados hasta ahora. Como se puede observar en las figuras 42, 43 y 44 los pulsos son muy erráticos, además de asimétricos y a veces muy breves, insuficientes para ser detectados como un flanco por el microcontrolador. Además, se pierde la secuencia de estados que permite detectar el sentido de giro como se describe en la figura 1.

Esto causa que el microcontrolador, que lee los pulsos mediante flancos, no sea capaz de leer todos y que por tanto no siga de forma correcta la cuenta de los mismos. Esto a su vez ocasiona que nuestra referencia de posición es errónea y en consecuencia nuestra onda sinusoidal de FEM no se encuentre estacionaria con la señal de posición, ya que no está sincronizada correctamente con el giro del rotor tal como lo percibe el microcontrolador. Además de que la comparación con el siguiente pulso es imposible, ya que no se solapan correctamente, impidiendo la correcta detección del sentido de giro (ver 2.4.4).

En la figura 37, aunque sí daba la sensación de ser correcto el envío de pulsos y el paso por 0 esto no es del todo cierto, ya que el pulso se mandaba con respecto al paso por 0, contando 500 pulsos (la mitad de los 1000 configurados) para así generar la señal. Esto ocasiona que, de perder pulsos, la pérdida es demasiado pequeña como para apreciarse, pero que, comparada con la FEM, sí se hace apreciable que no están completamente sincronizados.

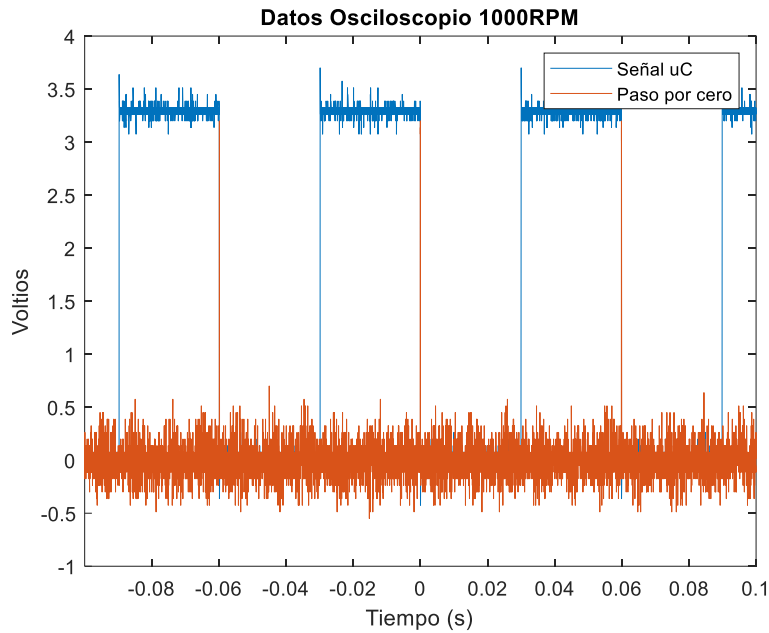
Para subsanar el problema hemos de ser conscientes de donde viene el mismo. En este caso es sencillo, es un problema de tiempo de respuesta del optoacoplador utilizado. Sabemos que es un problema de tiempo de respuesta porque se puede observar que, a bajas velocidades, el funcionamiento es correcto y que el problema aparece para frecuencias altas. En nuestro caso,

queremos un límite de funcionamiento de 1000 RPM y 1000 pulsos/rev, esto significa una frecuencia de pulsos de unos 16.6kHz. Para la solución buscaremos otro optoacoplador capaz de funcionar a 20kHz. Nuestro cálculo inicial puede haber sido erróneo debido a que tomamos los datos de los tiempos de respuesta de la hoja de datos del componente, sin embargo, estos datos proporcionados corresponden a una corriente de colector de 2mA, y nosotros estamos utilizándolo con una corriente de colector cercana a los 8mA. Esto no parecía un problema, ya que, según el fabricante, a menos resistencia de carga (más corriente) el tiempo debía ser cada vez más pequeño. Sin embargo, esto no es así, y es que puede que al habernos salido de las especificaciones marcadas por el fabricante las pruebas realizadas por el mismo no sean válidas con nuestro montaje, o quizás simplemente nuestra unidad es defectuosa. Sea como fuere decidimos optar esta vez por un optoacoplador de nivel lógico con Schmitt trigger.

Nuevamente nos encontramos con un problema: la disponibilidad. Debido a la falta de componentes electrónicos, el optoacoplador que pensábamos utilizar (un H11L1) no estaba disponible, o si lo estaba requería de pedidos superiores a 50 unidades, ilógico para las 3 que vamos a utilizar. Es por esto que decidimos utilizar un gate-driver para mosfets con optoacoplador, el J312. Este integrado lo teníamos disponible en el laboratorio y, realmente realiza la función de optoacoplador, pero requiere de una tensión de funcionamiento de entre 30 y 15V. En nuestro caso lo alimentaremos a 15V vía una fuente de alimentación externa, y la salida será dividida mediante un divisor resistivo hasta 3.3V para ser procesada por el microcontrolador.

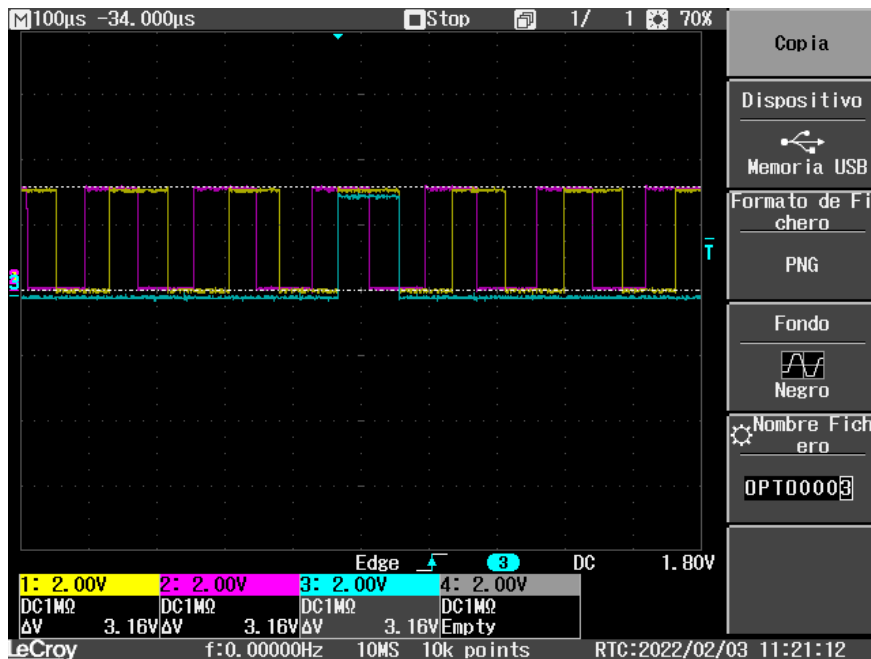
Además, para reducir el componente del ruido decidimos trabajar con la señal diferencial PAO y /PAO equivalente para cada señal. Esto fue debido a que, durante ensayos en el laboratorio observamos que, en el equipo, la señal de tierra del servopack y la señal SG (signal ground) de las señales de posición coincidían, lo que ocasionaba mucho ruido. Para evitar utilizar la señal de tierra decidimos utilizar la señal diferencial. La configuración con el nuevo optoacoplador es la misma, con una demanda de corriente a las señales del servopack de 10mA. Tras este cambio y tras la instalación de los 3 pertinentes optoacopladores en el banco de pruebas observamos estos datos:





**Figura 41 Oscilograma de la señal de paso por cero junto a la señal de posición generada por el microcontrolador (elaboración propia)**

Se puede ver cómo, ahora también tenemos sincronía a simple vista, pero veamos como quedan las señales A, B y Paso por 0



**Figura 42 Oscilograma de las señales A (ch1), B (ch2) y de paso por cero (ch3) tomada en el osciloscopio (elaboración propia)**

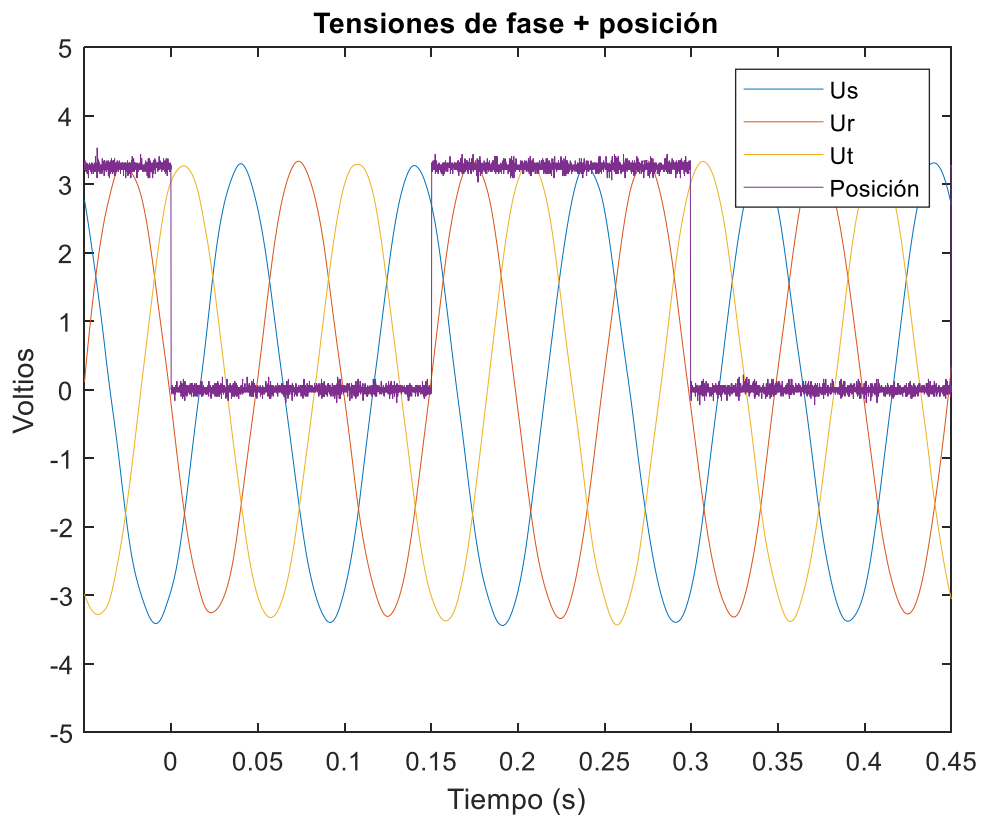
Vemos que ahora la señal es perfecta, incluso para los 16.6kHz máximos a los que vamos a trabajar, esto causa que ahora sí, el microcontrolador sea perfectamente capaz de leer los 1000 pulsos configurados en el Servopack. Esto lo obtenemos del puerto serie del programa IDE de Arduino.



**Figura 43 Captura de la salida por puerto serie de la aplicación Arduino IDE indicando los pulsos contados por revolución (elaboración propia)**

Efectivamente, ahora la lectura de pulsos es exacta, y por tanto la sincronía también lo será:

Una vez realizado el diseño de la realimentación de posición, en el siguiente capítulo se analizará el funcionamiento del servomotor y la implementación de su control mediante Simulink, lo que nos servirá de guía para la implementación definitiva mediante el microcontrolador.



**Figura 44 Oscilograma de las tensiones de cada una de las fases R, S, T y de la señal de posición generada por el microcontrolador tratadas con Matlab (elaboración propia)**

Una vez realizado el diseño de la realimentación de posición, en el siguiente capítulo se analizará el funcionamiento del servomotor y la implementación de su control mediante Simulink, lo que nos servirá de guía para la implementación definitiva mediante el microcontrolador.

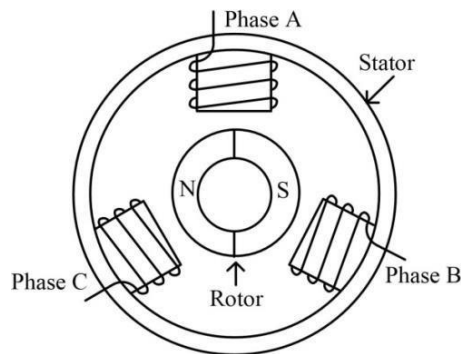
## 3. DISEÑO DEL SISTEMA DE CONTROL

### 3.1. INTRODUCCIÓN

A lo largo de este tercer capítulo se procederá al diseño del sistema de control tanto de corriente como de velocidad. Inicialmente realizaremos un análisis teórico del funcionamiento del motor para posteriormente comenzar con el diseño del control. Realizaremos un diseño teórico del propio motor síncrono de imanes permanentes y posteriormente de su controlador, junto con la realimentación de posición, velocidad y corriente necesaria. Una vez completado el diseño teórico se realizarán las pruebas de lectura de corriente y calibración y se implementará finalmente el controlador de forma digital en el microcontrolador de la tarjeta de control.

### 3.2. ANÁLISIS DE UN MOTOR SÍNCRONO DE IMANES PERMANENTES

Empecemos por comentar cómo está construido un motor síncrono de imanes permanentes. Como cualquier motor, las partes básicas que lo componen son el estator y el rotor. Al tratarse de un motor de imanes permanentes el rotor está compuesto por imanes con una polaridad fija (de ahí el nombre de permanentes). El estator por el contrario está formado por un bobinado de espiras trifásicas que, al ser excitadas, crearán un campo magnético.



**Figura 45 Esquema de las partes de un motor de imanes permanentes (Diagram)**

Para facilitar la comprensión, asumiremos un motor de  $p=1$ , es decir que solo tiene un par de polos.

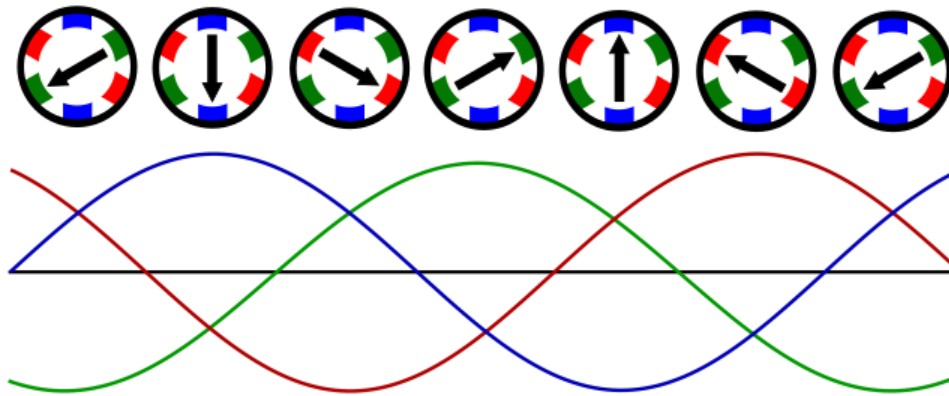
Para alguien sin conocimiento previo sobre motores eléctricos trifásicos el propio uso de este sistema trifásico puede resultar extraño y difícil de comprender. El uso de las tres fases para el movimiento rotativo de máquinas eléctricas no es casualidad, viene ligado fuertemente también al uso de corriente alterna. Como ya sabemos, cuando una corriente circula a través de un conductor, crea un campo magnético a su alrededor. Si la corriente utilizada para excitar la bobina es senoidal, el campo magnético creado por esta misma también lo será en su amplitud, aunque su orientación quedará siempre asociada a la disposición u orientación de la bobina. Así en la figura siguiente la fase azul produce campo en dirección vertical mientras que las fases verde y roja producen campo girado  $\pm 120^\circ$  respecto de esa orientación.

De esta forma el campo magnético alternará entre dos sentidos asociados al sentido de la corriente alterna en la bobina:



**Figura 46 Esquema de posición del eje con respecto a los diferentes polos (Diagram)**

Esto ocurre cuando tenemos solo una corriente senoidal, pero si al esquema del motor, con sus tres fases separadas 120 grados le aplicamos a cada una de esas fases una corriente senoidal, separadas entre ellas también 120 grados (esta vez eléctricos), observamos un fenómeno curioso:



**Figura 47 Secuencia de giro del eje con respecto a la forma de onda del campo magnético (Diagram)**

Si seguimos de izquierda a derecha el sentido de las flechas, que marcan el sentido del campo magnético observamos que éste, para un observador fijo con el estator, parece girar con el tiempo. Es en esta propiedad en la que se basa el accionamiento de las máquinas trifásicas rotativas.

Volviendo atrás, observamos que tenemos un estator en el que somos capaces de crear un campo magnético rotativo, y un rotor en el que existe permanentemente un campo magnético fijo con respecto al mismo. Si introducimos el rotor en el estator es intuitivo pensar que, al haber en el espacio dos campos magnéticos en posiciones diferentes, estos trataran de atraerse mutuamente. En este caso el campo magnético del rotor es capaz de moverse para alinearse con el campo magnético creado por el estator. Como el campo magnético del estator está rotando, el campo magnético del rotor también rotará, moviendo a su vez el rotor del motor. Hemos creado por tanto un motor síncrono de imanes permanentes.

Por supuesto lo mencionado anteriormente es simplemente una noción básica, pero es suficiente para comprender de forma básica el funcionamiento de este tipo de motores.

Conocido el funcionamiento básico del giro del motor, podemos pasar a modelizar este comportamiento con ecuaciones.

Empecemos primero con las tres corrientes sinusoidales separadas 120 grados entre ellas:

$$\begin{aligned}i_R &= I_p \cdot \cos(\omega t) \\i_S &= I_p \cdot \cos(\omega t - 120^\circ) \\i_T &= I_p \cdot \cos(\omega t - 240^\circ)\end{aligned}$$

Sabemos que el campo magnético está relacionado proporcionalmente con la corriente a través de la inductancia (mediante la ley de Faraday)

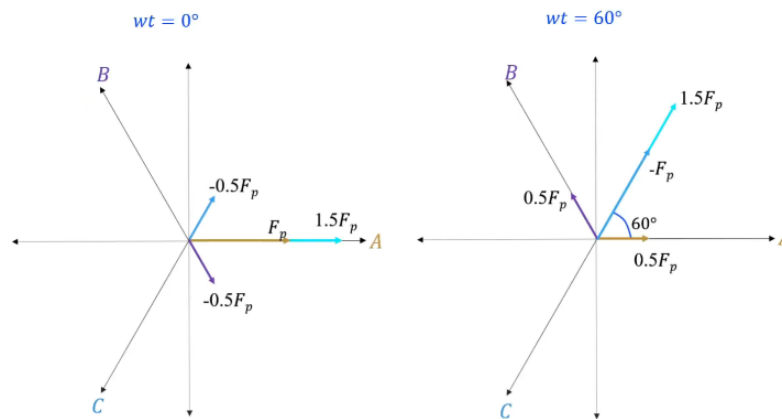
$$\Psi = L \cdot I$$

Luego podemos obtener tres ecuaciones también sinusoidales para las fuerzas magnetomotrices:

$$\begin{aligned}F_R &= F_p \cdot \cos(\omega t) \\F_S &= F_p \cdot \cos(\omega t - 120^\circ) \\F_T &= F_p \cdot \cos(\omega t - 240^\circ)\end{aligned}$$

Al estar superpuestas en el espacio la fuerza resultante de la combinación de las tres fuerzas magnetomotrices  $F_{net} = F_R + F_S + F_T = 1.5 \cdot F_p \cdot \cos(\omega t)$

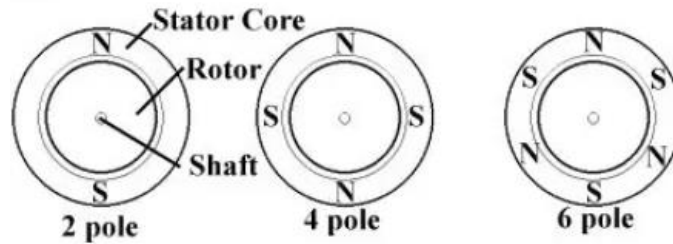
Para visualizar la suma más claramente podemos optar por la representación vectorial de estas funciones, de esta forma observamos que  $F_{net}$  es un vector que gira con velocidad angular  $\omega$  en el estator:



**Figura 48 Esquema vectorial (Lerma)**

El contar con estas ecuaciones nos permitirá, por ejemplo, conocer la velocidad de giro del rotor en base a la frecuencia de las corrientes trifásicas aplicadas. Ya que recordemos que nuestra meta es realizar un control eléctrico del motor con el que aplicar acciones físicas.

Para poder calcular la velocidad angular de giro del rotor en base a la frecuencia primero debemos dejar clara la diferencia entre ángulos eléctricos y mecánicos. El ángulo mecánico es el ángulo de giro del rotor físico con respecto del estator. Está claro que este ángulo se recorre de 0 a 360 grados en una revolución del rotor. Sin embargo, esto no es necesariamente igual para el ángulo eléctrico. Digamos que el ángulo eléctrico es (con respecto al mecánico), cuantos grados se pueden “contar” desde que sales de un polo hasta que vuelves a encontrarte con el mismo polo. Veamos un ejemplo para entenderlo apropiadamente:



**Figura 49 Esquema de ejemplo de motores con distintos números de polos (Diagram)**

En la primera imagen de la izquierda observamos que, desde que salimos del polo N podemos contar 360 grados (del círculo gráfico) hasta que nos volvemos a encontrar con otro polo N. Decimos pues que tiene 1 par de polos (par, ya que tenemos el polo norte y el polo sur). Sin embargo, en la segunda imagen ya no pasa esto. Solo nos podemos recorrer 180 grados tras salir de un polo hasta encontrar otro polo igual. Es decir que por cada 180 grados mecánicos hay 360 grados eléctricos. Observamos que hay 2 pares de polos en la segunda imagen. Y de forma idéntica observamos 3 pares de polos en la tercera imagen y 120 grados mecánicos por cada 360 grados eléctricos.

Que haya 180 grados mecánicos por cada 360 grados eléctricos quiere decir que, cuando el vector “eléctrico” haya girado 360 grados, el vector “mecánico” (que es el propio eje del rotor), solo habrá girado 180 grados. Además, se ve claramente que 1 par de polos equivale a 360 grados mecánicos por cada grado eléctrico, 2 pares de polos a 180, 3 a 120, etc.

Decimos por tanto que:

$$\omega_{elec} = p \cdot \omega_{mec}$$

Siendo  $p$  el número de pares de polos. Si la velocidad angular del rotor en RPM la nombramos como:

$$N_s = \frac{rev}{min}$$

Está claro que su equivalente en rad/s:

$$\omega_{mec} = N_s \cdot \frac{2\pi}{60}$$

Si ahora sustituimos en la primera ecuación  $\omega_{mec}$  por la tercera ecuación obtenemos:

$$\omega_{elec} = p \cdot N_s \cdot \frac{2\pi}{60}$$

Si despejamos  $N_s$  en función de  $\omega_{elec}$  nos queda la velocidad angular del rotor en RPM en función de la velocidad angular eléctrica:

$$N_s = \omega_{elec} \cdot \frac{60}{p \cdot 2\pi}$$

Y si quisiéramos expresar la velocidad mecánica en función de la frecuencia de las corrientes eléctricas, recordando que:

$$\omega = 2\pi f$$

Nos quedaría que:

$$N_s = \frac{60 \cdot f}{p}$$

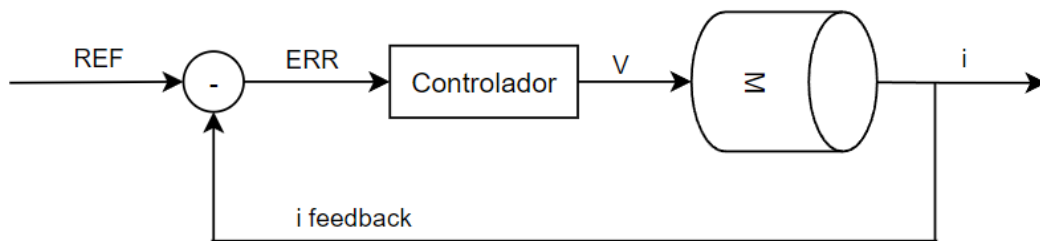
Esto denota que el conocimiento matemático del funcionamiento del motor es imprescindible para su control. Y es de hecho ese mismo el siguiente tema a tratar, cómo realizar el control del motor de imanes permanentes de forma que podamos ajustar la corriente que consume, el par que aporta o la velocidad angular a la que gira. Para ello nos valdremos del control por orientación de campo, cuyo nombre ya nos da una pista de su funcionamiento.

### 3.3. CONTROL DEL MOTOR DE IMANES PERMANENTES

En este capítulo nos centraremos en entender el control del motor de imanes permanentes para poder posteriormente implementarlo en Simulink y finalmente de forma discreta con un microcontrolador.

Para el control del motor lo que buscamos es un control sobre el campo magnético a aplicar en el núcleo del estator. Como hemos comentado repetidas veces anteriormente, el campo magnético es proporcional a la corriente. Es por esto que para poder ejercer cualquier tipo de control sobre el motor debemos de ser capaces de controlar fielmente el vector de corriente giratorio que le estamos aplicando al estator.

Empecemos por tratar de controlar la corriente que inyectamos en el mismo, para después utilizar la comanda de corriente para controlar el par y la velocidad angular del motor. Si dibujáramos en un diagrama de bloques lo que queremos obtener quedaría algo así:

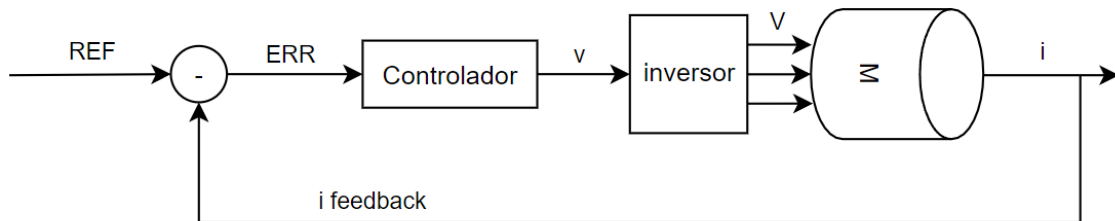


**Figura 50 Esquema de control del motor simplificado (elaboración propia)**

Aplicaríamos una referencia de corriente, a la que restando la corriente que leamos como feedback entregaremos el error al controlador. El controlador a su vez generaría una señal de control en forma de tensión (ya que nosotros podemos interactuar con el motor mediante la tensión que le apliquemos a sus 3 fases).

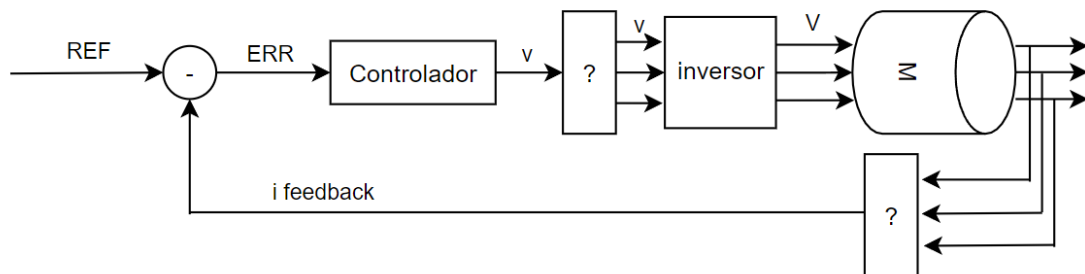
Este diagrama es sencillo, pero sirve para dejar claro cuál es nuestro objetivo. Sin embargo, nos muestra varios problemas clave. El primer problema es cómo puede nuestro controlador interactuar con el motor, es decir, cuál debe ser la etapa de potencia a interponer entre el controlador y el motor para poder entregarle al mismo la potencia que requiera. El segundo problema es que, como hemos visto al principio en la parte de explicación teórica, nuestro motor funciona con tensiones (y corrientes) trifásicas que además cambian de forma sinusoidal con el tiempo.

Más adelante entraremos en una explicación más en detalle, pero comencemos por contestar a la primera duda. El circuito de potencia que sirve para interactuar entre el controlador y el motor es un inversor. El inversor recibirá nuestra tensión a comandar al motor y será capaz de aplicarla al mismo con la suficiente potencia como para poder energizar las bobinas del motor y generar el campo magnético que andamos buscando. No entraremos ahora en el funcionamiento concreto del mismo, lo haremos más adelante. Sin embargo, vamos a ir actualizando nuestro diagrama para seguir el hilo de lo que vamos haciendo:



**Figura 51 Esquema de control del motor simplificado con inversor (elaboración propia)**

Con el diagrama actualizado podemos ahora tratar de resolver el segundo problema propuesto. El motor es una máquina trifásica, que por tanto recibe una tensión trifásica que generará corrientes trifásicas. El inversor recordemos que es un simple intermediario entre la tensión de control generada por el controlador y el motor, por lo que este también tendrá que recibir una tensión trifásica para poder comunicarla al motor. Lo que nosotros buscamos es algún método para poder trabajar con valores escalares y poder después transformarlo a valores trifásicos giratorios en el tiempo. Para aclararlo volvamos una vez más a actualizar nuestro diagrama de control para ver que pieza nos falta:

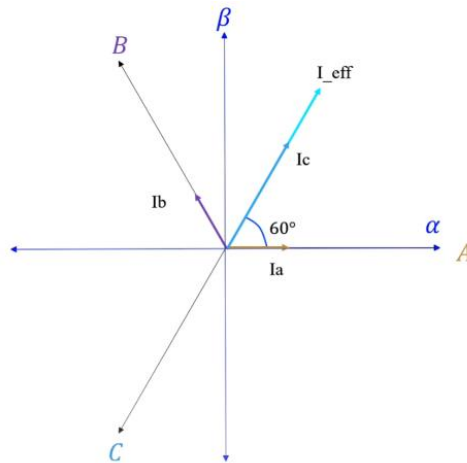


**Figura 52 Esquema de control del motor simplificado con inversor y transformadas vectoriales (elaboración propia)**

Igual que hemos hecho anteriormente tratemos de expresar nuestro problema en forma matemática para poder buscar una solución y ampliar nuestro conocimiento.

Cómo vimos anteriormente las tres corrientes de las fases se suman para generar un vector:





**Figura 53 Esquema vectorial de corrientes en ejes alfa y beta (Lerma)**

Nótese que estamos combinando 3 vectores en el plano para generar un cuarto vector combinación lineal de los anteriores. Al encontrarnos en un espacio 2-dimensional no necesitamos de una combinación lineal de 3 vectores para generar un cuarto, realmente nos bastaría con 2. En la imagen anterior se han indicado los ejes A, B y C, pero también se han indicado los ejes  $\alpha$  y  $\beta$ . Estos ejes  $\alpha$  y  $\beta$  son suficientes para generar con una combinación lineal de ellos cualquier vector en el espacio 2-dimensional. Para pasar de un sistema de 3 vectores a uno de 2 podemos representar los vectores A, B y C como combinación lineal de  $\alpha$  y  $\beta$  y despejar estos para obtener la transformada:

$$i_{\alpha} = \frac{2}{3} [i_A - 0.5i_B - 0.5i_C]$$

$$i_{\beta} = \frac{2}{3} \left[ \frac{\sqrt{3}}{2} i_B - \frac{\sqrt{3}}{2} i_C \right]$$

Si lo expresamos en forma matricial nos queda que:

$$\begin{bmatrix} i_{\alpha} \\ i_{\beta} \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} 1 & -0.5 & -0.5 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix}$$

A esta transformada se le conoce como la transformada de Clarke y es el primer paso para poder transformar nuestro sistema de 3 ecuaciones rotativas en un sistema escalar estático.

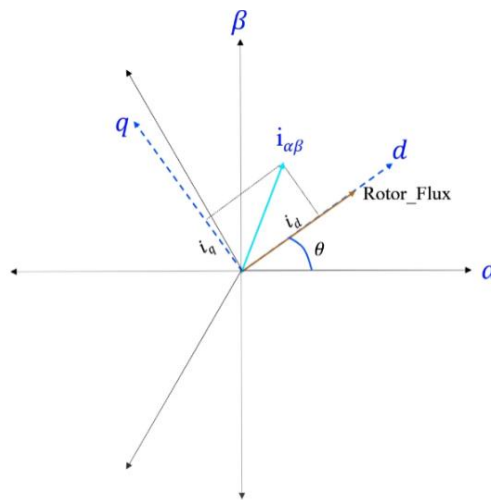
Veamos ahora cual es el segundo paso.

Nos encontramos pues con el siguiente problema: a nosotros nos gustaría poder controlar la posición relativa del vector de campo magnético que creamos con el estator con referencia al vector de campo magnético creado por el propio imán permanente del rotor. Esto sin embargo resulta bastante complicado teniendo en cuenta que el vector de campo magnético del rotor está continuamente girando cuando la máquina está en funcionamiento, y por lo tanto el posicionamiento relativo entre el vector que nosotros podemos generar y éste se complica. Sería mucho más útil poder tratar ambos vectores en unos ejes de referencia definidos de forma que el ángulo entre ellos fuera constante durante el tiempo en régimen permanente representando la tendencia continua del imán del rotor a alcanzar al imán del estator sin conseguirlo.

Lo comentado anteriormente es posible si se conoce el ángulo girado por el rotor: en nuestro caso contamos de un sensor óptico en el eje, luego la implementación de este sistema es viable, veamos cómo.

Si queremos crear un sistema de referencia tal que se mantenga fijo con el vector de campo magnético del rotor lo razonable sería hacer coincidir uno de los vectores base del nuevo sistema con el vector campo magnético, de esta forma será mucho más simple realizar la transformación de bases.

Para realizar la transformación de una base a otra bastará con expresar la nueva base fija con el vector de campo magnético del rotor en función de los vectores de la antigua base formada por  $\alpha$  y  $\beta$ . De forma gráfica lo que queremos hacer es esto mismo:



**Figura 54 Esquema vectorial de corrientes en ejes d y q (Lerma)**

Como vemos, si realizamos la transformación de las bases con respecto a  $\theta$ , siendo este el ángulo del rotor y por tanto del vector de campo magnético del mismo, podemos obtener un sistema de referencia que esté "ligado" al propio eje magnético. A este nuevo sistema de referencia le llamaremos DQ siendo el eje del alineado con el flujo magnético del rotor y el eje q el perpendicular al mismo.

Para obtener esta transformada entre bases bastará con proyectar el vector formado por una combinación lineal de los vectores  $\alpha$  y  $\beta$  sobre los nuevos ejes d y q. Quedando de la siguiente manera:

$$i_d = i_\alpha \cos(\theta) + i_\beta \sin(\theta)$$

$$i_q = i_\alpha - \sin(\theta) + i_\beta \cos(\theta)$$

Que puede ser expresado en forma matricial como:

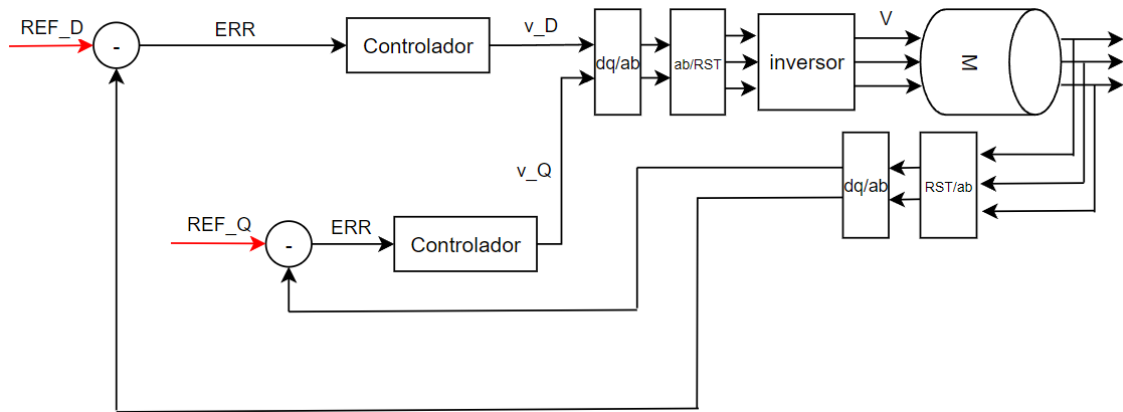
$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}$$

A esta transformación entre bases se le conoce como la transformada de Park.

Lo que hemos conseguido es una base sobre la que trabajar en la que los vectores de flujo de campo magnético tanto del estator como del rotor aparecen como estáticos en régimen permanente.

Si juntamos ambas transformadas podemos observar cómo podemos transformar una serie de 3 vectores trifásicos que rotan en el espacio en 2 vectores estáticos referenciados a la posición del rotor.

Esto nos permite trabajar en los ejes d y q con valores escalares y ya no vectoriales, y además de valores fijos en régimen permanente, para la fácil implementación con un controlador clásico de tipo PI. Sin embargo, nótese que no es exactamente lo que buscábamos. Nosotros buscábamos un solo valor escalar, de forma que nuestro controlador solo atienda a una variable. Esto es fácil de solucionar, simplemente añadimos un segundo controlador, de forma que tengamos un controlador por cada variable a controlar. Para dejar este concepto más afianzado actualicemos nuestro diagrama de bloques para ver cómo queda:



**Figura 55 Esquema de control del motor completo con referencias (elaboración propia)**

Como podemos observar tenemos dos referencias ya que el vector de flujo magnético (y por tanto el de corriente) en los ejes d y q necesita de dos valores para definirse, uno para el valor de d y otro para el valor de q. Esto es exactamente igual que cuando definimos un vector en la base formada por ejes X e Y, formado de esta manera  $V = (x, y)$ . Ahora ya tenemos un esquema de cómo va a trabajar nuestro control por corriente. Sin embargo, ahora querríamos modelizar el comportamiento del control por bucle cerrado y para ello necesitamos saber cómo se comporta el motor expresando el comportamiento en forma matemática. Para ello adentrémonos un poco más a fondo en la descripción matemática del funcionamiento del motor.

### 3.4. ECUACIONES DEL MOTOR DE IMANES PERMANENTES

Empecemos primero por recordar la ley de Faraday:

$$U = \frac{d\Psi}{dt}$$

Y por otra parte también sabemos que:

$$U = L \frac{di}{dt}$$

Por tanto, está claro que:

$$\frac{d\Psi}{dt} = L \frac{di}{dt} \rightarrow \Psi = Li$$

Esto también se aplica al espacio vectorial:

$$\vec{\Psi} = L \cdot \vec{i}$$

Y recordemos que podemos expresar un vector en forma polar de esta forma:

$$\vec{\Psi} = |\vec{\Psi}| e^{j\theta}$$

Esto nos será muy útil más adelante. Continuemos ahora pensando qué partes comprenden el circuito de la máquina completa.

Tenemos el flujo del estator, que llamaremos  $\Psi_s$ . También tenemos el flujo del rotor, que llamaremos  $\Psi_r$ . Y por tanto el flujo total  $\Psi_T = \Psi_s + \Psi_r$ .

Además del flujo magnético de la máquina también tenemos que tener en cuenta la resistencia de los cables.

Por lo tanto, recapitulando tenemos dos caídas de tensión en la máquina rotativa:

$$\vec{U}_{FLUJO} = \frac{d\vec{\Psi}_T}{dt} \quad \& \quad \vec{U}_{CABLE} = R\vec{i}_s$$

También podemos describir la caída de tensión del flujo (recordando que  $\frac{d\vec{\Psi}_s}{dt} = L \frac{d\vec{i}_s}{dt}$ ) como:

$$\vec{U}_{FLUJO} = L \frac{d\vec{i}_s}{dt} + \frac{d\vec{\Psi}_r}{dt}$$

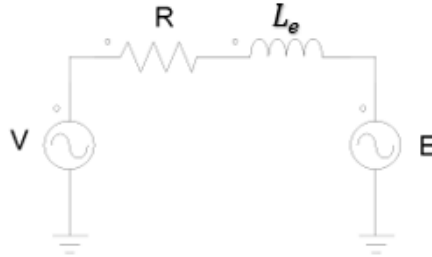
Y podemos llamar a  $\frac{d\vec{\Psi}_r}{dt}$  como  $\vec{E}_s$ , siendo esta la tensión que observaríamos si midiéramos la tensión sin excitar las bobinas a la vez que hacemos girar el rotor, normalmente conocida como Back EMF.

Recapitulando, la ecuación “vector espacial” de la máquina y con vector espacial nos referimos a que no hace referencia a ningún “circuito” en concreto como podría ser el circuito de alguna de las tres fases, sino que al ser vectorial hace referencia a la totalidad de la máquina. Esto se vuelve más fácil de comprender si completamos la ecuación:

$$\vec{V} = R\vec{i}_s + L_s \frac{d\vec{i}_s}{dt} + \vec{E}_s$$

Podemos entender la tensión, no como una tensión en alguna de las fases, sino como el propio vector espacial suma de las tres tensiones de fases distintas. Esto es muy útil en este caso, ya que nos permite definir un “circuito” único para el funcionamiento completo de la máquina.

Podemos imaginarnos este “circuito” como:



**Figura 56 Representación esquemática del circuito eléctrico de las fases del motor (elaboración propia)**

Bien, sin embargo, volvemos a tener el problema anterior: no queremos trabajar con ecuaciones vectoriales, lo que queremos hacer es trabajar con las partes escalares de estas ecuaciones. Hacer esto no es tan inmediato como podría parecer ya que dentro de la ecuación vectorial anteriormente mostrada tenemos derivadas vectoriales. Para poder expresar lo que queremos hacer rigurosamente debemos realizar la expansión de estos términos. Hagámoslo término por término.

El primer término es muy sencillo, y es por esto que lo dejaremos para el final, empecemos con el último:

$$\vec{E}_s = \frac{d\vec{\Psi}_r}{dt}$$

Recordemos pues con este simple ejemplo cómo se deriva un vector. Para entender mejor qué es lo que vamos a hacer es útil expresar este vector de forma polar en el dominio del tiempo (igual que hemos hecho al comienzo de esta parte):

$$\frac{d\vec{\Psi}_r}{dt} = \frac{d(|\vec{\Psi}_r| \cdot e^{j\theta})}{dt}$$

En este caso, como vimos al principio, la posición de este vector no es constante. Es decir,  $\theta$  no es un ángulo constante, sino que al estar el vector rotando en el tiempo debemos expresarlo de esta manera  $\theta = \omega t$ . Por lo tanto, reescribamos la ecuación anterior:

$$\frac{d\vec{\Psi}_r}{dt} = \frac{d(|\vec{\Psi}_r| \cdot e^{j\omega t})}{dt}$$

Ahora ya se empieza a ver cómo podemos realizar la derivada de este vector, hagámoslo paso a paso:

$$\frac{d\vec{\Psi}_r}{dt} = \left( \frac{d|\vec{\Psi}_r|}{dt} + j\omega|\vec{\Psi}_r| \right) \cdot e^{j\omega t}$$

Es fácil seguir lo que hemos hecho pensando en la derivada de una multiplicación, en la que primero derivamos el primer término, multiplicado por el segundo sin derivar y luego le sumamos el segundo término derivado por el primero sin derivar.

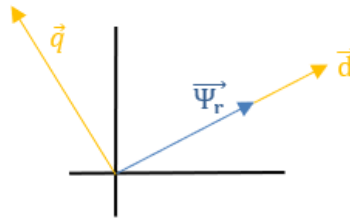
Aclarado esto podemos fijarnos en que el módulo del flujo del rotor, al venir de un imán permanente, es constante en módulo y por tanto su derivada es nula. Esto nos deja con la siguiente ecuación:

$$\frac{d\vec{\Psi}_r}{dt} = j\omega |\vec{\Psi}_r| e^{j\omega t}$$

Aprovechamos este momento antes de continuar para recordar que queremos trabajar en una base de coordenadas fija con el rotor. Esto quería decir que íbamos a dejar fijo uno de los ejes de esta base “quieta” unido al vector de flujo magnético. Esto quiere decir que lo que acabamos de calcular, expresado en nuestra base “quieta” unida al rotor quedaría expresado en forma cartesiana como:

$$\left(\frac{d\vec{\Psi}_r}{dt}\right)_{dq} = 0 + j\omega |\vec{\Psi}_r|$$

Esto es ya que el vector  $e^{j\omega t}$  unitario es el propio eje d de nuestra base fija.



**Figura 57 Representación vectorial del flujo del eje del motor sobre los ejes d y q (elaboración propia)**

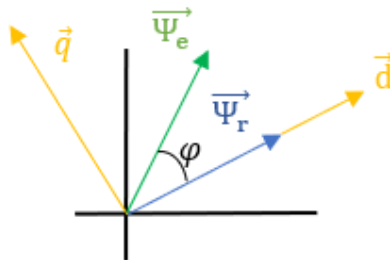
Habiendo dejado esto claro podemos continuar con los próximos términos de la ecuación:

$$L_e \frac{d\vec{i}_s}{dt}$$

Procederemos como hemos hecho con el anterior término expresándolo de forma polar y realizando la derivada de la misma manera que antes:

$$L_e \frac{d\vec{i}_s}{dt} = \left( L_e \frac{d|\vec{i}_s|}{dt} + j\omega L_e |\vec{i}_s| \right) e^{j(\omega t + \varphi)}$$

Nótese que ahora al vector unitario  $e^{j\omega t}$  se le ha añadido el término  $\varphi$ , este indica que hay un cierto desfase (a priori desconocido) entre el vector de flujo de campo magnético del rotor y el vector de corriente, que recordemos está alineado con el vector de flujo magnético del estator. Para que quede más claro podemos ilustrarlo en un diagrama vectorial:



**Figura 58 Representación vectorial del flujo magnético del estator y del eje sobre los ejes d y q (elaboración propia)**

Esta vez ya no podemos asegurar que la derivada del módulo de la corriente sea nula ya que no tiene por qué serlo.

Por último, el primer término de la ecuación, que antes habíamos dejado para el final es el más simple. Expresado en forma vectorial polar quedará como:

$$R|\vec{i}_s|e^{j(\omega t + \varphi)}$$

Y si queremos expresar el vector tensión también en forma polar nos quedará como:

$$\vec{V} = |\vec{V}|e^{j(\omega t + \varphi')}$$

Ese  $\varphi'$  hace referencia a un ángulo distinto al de la corriente pero que denota que no tiene ninguna alineación especial con alguno de los vectores que forman la base, y que por tanto se debe expresar de forma genérica en la ecuación final. Vamos pues a ver como quedaría todo lo anotado anteriormente junto en una única ecuación vector espacial de la máquina.

$$|\vec{V}|e^{j(\omega t + \varphi')} = R|\vec{i}_s|e^{j(\omega t + \varphi)} + L_e \frac{d|\vec{i}_s|}{dt} e^{j(\omega t + \varphi)} + j\omega L_e |\vec{i}_s| e^{j(\omega t + \varphi)} + j\omega |\vec{\Psi}_r| e^{j(\omega t)}$$

Si ahora expresamos esta ecuación vector espacial en forma cartesiana sobre la base “unida” al vector de flujo magnético del rotor nos quedaría algo así:

$$v_d + jv_q = R(i_d + ji_q) + L_e \left( \frac{di_d}{dt} + j \frac{di_q}{dt} \right) + j\omega L_e (i_d + ji_q) + j\omega |\vec{\Psi}_r|$$

Como vemos, el último término (asociado al vector unitario  $e^{j(\omega t)}$ ) no se desdobra en 2 partes referidas a d y a q, ya que ya sabemos que ese mismo unitario es el propio eje d.

Si ahora separamos los términos de la ecuación nos queda que:

$$v_d = Ri_d + L_e \frac{di_d}{dt} - \omega L_e i_q$$

$$v_q = Ri_q + L_e \frac{di_q}{dt} + \omega L_e i_d + \omega |\vec{\Psi}_r|$$

Veamos ahora cómo podemos obtener el par de la máquina.

El par de la máquina viene definido por esta expresión:

$$T = \frac{3}{2} \cdot p (\vec{\Psi}_t \wedge \vec{i}_s)$$

Y recordemos que:

$$\vec{\Psi}_t = \vec{\Psi}_s + \vec{\Psi}_r = L_s \vec{i}_s + \vec{\Psi}_r$$

Si incorporamos esta última expresión en la anterior y desarrollamos el producto vectorial nos encontramos con que:

$$\vec{\Psi}_t \wedge \vec{i}_s = L_s \vec{i}_s \wedge \vec{i}_s + \vec{\Psi}_r \wedge \vec{i}_s$$

Y recordando las propiedades del producto vectorial obtenemos que:

$$L_s \vec{i}_s \wedge \vec{i}_s = 0$$

Y que, por lo tanto:

$$\vec{\Psi}_t \wedge \vec{i}_s = \vec{\Psi}_r \wedge \vec{i}_s$$

$$T = \frac{3}{2} \cdot p (\vec{\Psi}_r \wedge \vec{i}_s)$$

Esto nos indica que para maximizar el par la corriente debe de ser perpendicular al vector de flujo magnético del estator. Algo que intuitivamente parece tener sentido, dos imanes separados entre ellos dispuestos perpendicularmente hacen más fuerza para unirse que si su posición es distinta de la perpendicular.

Si realizamos el cálculo de la anterior ecuación en la base formada por los ejes d y q nos queda que:

$$T = \frac{3}{2} p \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ |\vec{\Psi}_r| & 0 & 0 \\ i_d & i_q & 0 \end{vmatrix}$$

$$= \frac{3}{2} p (|\vec{\Psi}_r| i_q)$$

Si lo hacemos en cualquier otra base obtenemos que:

$$T = \frac{3}{2} \cdot p \cdot |\vec{\Psi}_r| \cdot |\vec{i}_s| \cdot \sin(\lambda - \theta)$$

Siendo  $\lambda - \theta$  el ángulo entre ambos vectores.

Además del par, tenemos otra ecuación mecánica de la máquina rotativa, la relacionada con la velocidad angular y el par:

$$T_{mot} - T_{carga} = J \frac{d^2\theta}{dt^2}$$

Dicho esto, ya tenemos una serie de ecuaciones escalares que nos podrían servir, por ejemplo, para simular el comportamiento del sistema o calcular su función de transferencia nosotros mismos. Para expresarlas despejaremos  $i_d$  e  $i_q$  de sus respectivas ecuaciones de tensión:

$$\frac{di_d}{dt} = \frac{1}{L_e} [v_d - Ri_d + \omega L_e i_q]$$

$$\frac{di_q}{dt} = \frac{1}{L_e} [v_q - Ri_q + \omega L_e i_d - \omega |\vec{\Psi}_r|]$$

$$T = \frac{3}{2} p (|\vec{\Psi}_r| i_q)$$

$$\frac{d\omega}{dt} = \frac{1}{J} (T - T_{carga})$$

Sin embargo, dado que a día de hoy tenemos herramientas tan potentes como Matlab o Simulink, podemos trabajar con ecuaciones vectoriales dentro de estas herramientas, de forma que comprender qué es lo que está pasando sea más sencillo.

Empecemos pues, desde 0 en el siguiente epígrafe con un modelo de Simulink en el que comenzar a implementar nuestras ecuaciones.



### 3.5. MODELADO DE LA MÁQUINA CON SIMULINK

Antes de comenzar expliquemos un par de conceptos más detalladamente, que harán nuestro trabajo más fácil en el futuro.

El primero es recordar que, como se explicó al inicio de la teoría, la velocidad angular del rotor y la velocidad angular eléctrica no tienen por qué ser la misma si el número de pares de polos no es 1. Esto quiere decir que para todas nuestras ecuaciones en las que tenemos en cuenta el ángulo o la velocidad de giro, esta será siempre la eléctrica, y que, como vamos a trabajar también con la mecánica, utilizaremos  $\omega$  para la velocidad angular mecánica y  $p \cdot \omega$  para la eléctrica.

Otro concepto importante es que, en lugar de utilizar  $|\vec{\Psi}_r| \cdot p$  utilizaremos  $k_e$ , y que en lugar de utilizar  $\frac{3}{2} |\vec{\Psi}_r| \cdot p$  utilizaremos  $k_t$ .

Además, tal y como se ha comentado en el apartado anterior haremos una simulación vectorial del comportamiento de la máquina. Esto quiere decir que, aunque tenemos claro que las tensiones y corrientes del motor real vienen expresadas como trifásicas, a nosotros solo nos hacen falta una combinación lineal de 2 vectores para expresarlas en el espacio.

Comencemos por la ecuación vectorial de la tensión:

$$\vec{V} = R\vec{i}_s + L_s \frac{d\vec{i}_s}{dt} + \vec{E}_s$$

Como

sabemos:

$$\vec{E}_s = \frac{d\vec{\Psi}_r}{dt} = j\omega p |\vec{\Psi}_r| e^{j(\omega t)} = \omega p |\vec{\Psi}_r| e^{j(\omega t + \frac{\pi}{2})} = k_e \omega e^{j(\omega t + \frac{\pi}{2})}$$

Nótese que podemos trabajar indistintamente en forma polar o cartesiana según nos sea más cómodo. Además, hemos substituido  $\omega$  por  $\omega p$  ya que, como se ha mencionado anteriormente, ahora hacemos distinción entre velocidad angular mecánica y eléctrica.

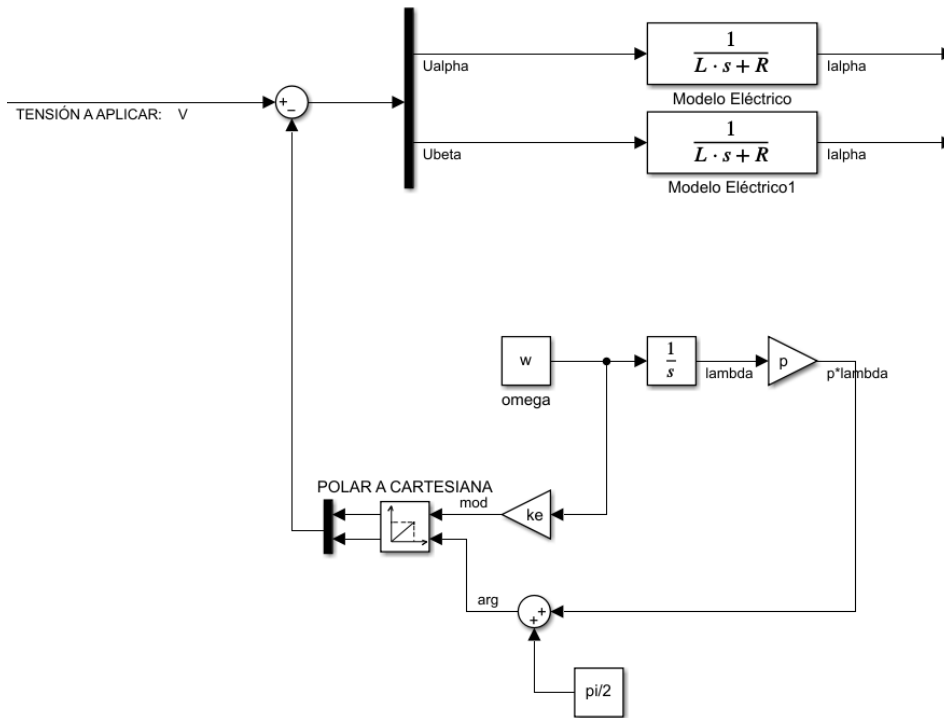
Lo que estamos buscando, como se ha visto al final del anterior capítulo, es una forma de obtener cuales son las corrientes para una tensión aplicada. Por esto mismo debemos despejar la corriente de nuestra ecuación vectorial. Sin embargo, uno de los términos de la corriente en nuestra ecuación está dentro de la derivada. Para poder trabajar algebraicamente con la ecuación podemos hacer la transformada a Laplace:

$$\vec{V} - k_e \omega e^{j(\omega t + \frac{\pi}{2})} = R\vec{i}_s + L_s \vec{i}_s s$$

Ahora ya podemos despejar la corriente de nuestra ecuación:

$$\left( \vec{V} - k_e \omega e^{j(\omega t + \frac{\pi}{2})} \right) \frac{1}{s \cdot L_s + R} = \vec{i}_s$$

Con esto ya podemos empezar a trabajar en Simulink. Creamos una nueva simulación e incluimos lo que acabamos de deducir:



**Figura 59 Modelado de las ecuaciones del motor en Simulink (corriente) (elaboración propia)**

Algo a comentar es que para obtener el ángulo del rotor a partir de la velocidad angular del mismo basta con integrarla.

Este será nuestro “bloque” de equivalencia eléctrica del motor, podemos ahora obtener el par a través de la corriente. Lo que haremos será incorporar:

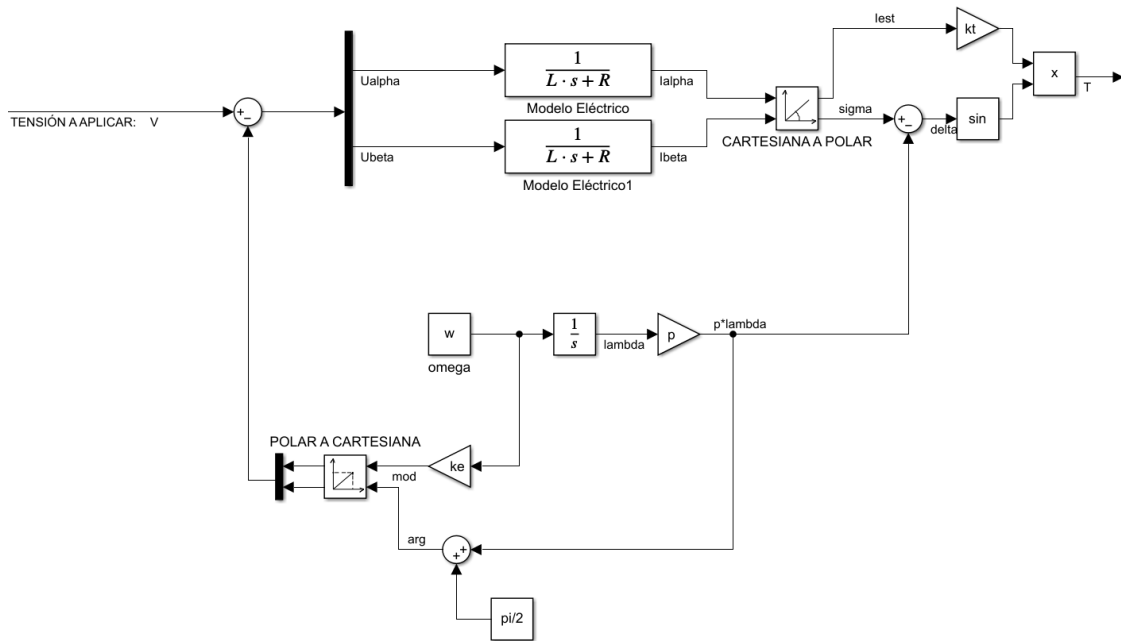
$$T = \frac{3}{2} \cdot p \cdot |\overline{\Psi}_r| \cdot |\vec{i}_s| \cdot \sin(\lambda - \theta)$$

A nuestro modelo, que reajustado a nuestras variables quedará tal que así:

$$T = k_p \cdot |\vec{i}_s| \cdot \sin(\varphi - \lambda \cdot p)$$

La diferencia en los ángulos es que ahora llamaremos  $\varphi$  al ángulo de la corriente y  $\lambda \cdot p$  al ángulo del vector flujo magnético del rotor. Además de esta forma se puede ver claramente como nuestros vectores están girando, y que cuando queramos implementar el control tendremos que “transformarlos” a nuestra base unida al rotor.

Añadiendo esta ecuación al modelo nos queda tal que así:

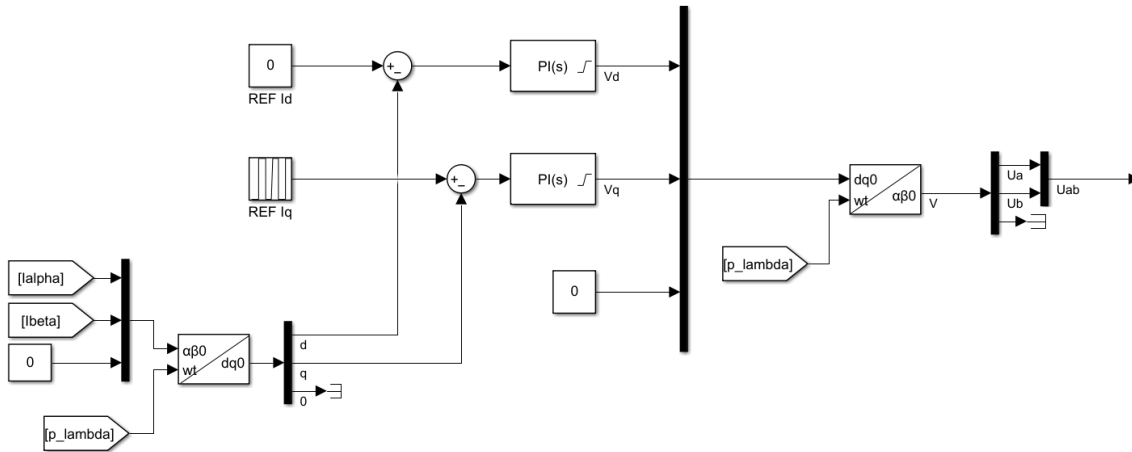


**Figura 60 Modelado de las ecuaciones del motor en Simulink (corriente y par) (elaboración propia)**

Bien, ahora ya tenemos el modelo eléctrico y mecánico de nuestro motor en ecuaciones vectoriales descrito en Simulink. Es buen momento para recordar el esquema que teníamos de nuestro controlador, representado en la figura 59. A falta de conocer cómo podemos implementar el inversor tratemos primero de implementar el control ya que, al tratarse de una simulación, a nuestro modelo no le importa si la señal de control que le aportamos es capaz de aportar la potencia necesaria para mover el motor. Más adelante sin embargo entraremos en el funcionamiento del inversor y en su implementación.

A cerca del control recordemos que lo que necesitamos son dos controladores, uno para la corriente en el eje d y otro para la corriente en el eje q. Por esto mismo también necesitaremos dos referencias. Además, como vamos a trabajar en la base DQ necesitaremos nuestras transformadas de Clarke y de Park para poder realizar la transformación, así como un feedback tanto de la posición del eje del rotor como de las corrientes que nuestro modelo eléctrico del motor nos entrega.

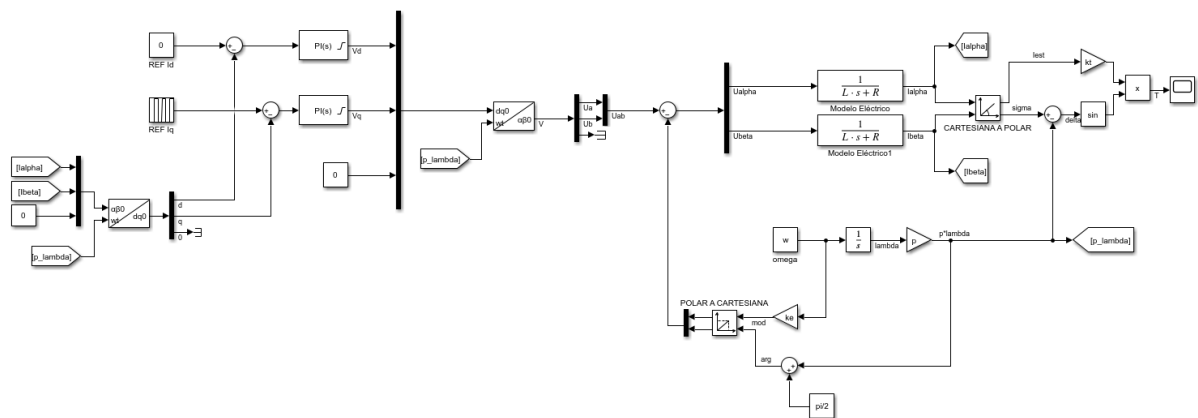
Veamos cómo se implementa esto mismo en Simulink:



**Figura 61 Implementación del control en Simulink con realimentación de corriente (elaboración propia)**

Como vemos, hemos implementado lo anteriormente mencionado. Hay un pequeño problema de fácil solución, el bloque de transformación de bases al no ser más que una multiplicación de matrices requiere de 3 entradas, y su inverso nos dará 3 salidas. Como solo nos interesan 2 de esas entradas y salidas, en la tercera entrada daremos un valor nulo e ignoraremos la tercera salida.

La salida  $U_{ab}$  nombrada en el esquema es nuestra salida de control, de forma que todo el sistema luce de esta manera:



**Figura 62 Esquema de control del motor completo en Simulink (elaboración propia)**

Podemos observar que solo hemos añadido la parte del control y que además no hemos tenido que modelar (todavía) el inversor, ya que como hemos comentado antes, al ser simplemente un modelo teórico, no tenemos por qué preocuparnos sobre la posibilidad de inyectar la potencia necesaria por parte de las señales de control, algo de lo que sí que nos tendremos que ocupar en la realidad.

Con el modelo preparado lo siguiente que necesitamos es ajustar nuestros controladores para poder obtener la respuesta que esperamos.

### 3.6. AJUSTE DEL CONTROL DE CORRIENTE

Cómo se ha comentado previamente necesitamos ahora asignar los valores a las componentes del controlador tipo PI.

Estas componentes serán la ganancia de la parte proporcional y la constante de tiempo integral. Para hacer esto debemos observar el sistema que vamos a controlar, para el cual ya tenemos sus ecuaciones:

$$\frac{di_d}{dt} = \frac{1}{L_e} [v_d - Ri_d + \omega L_e i_q]$$

$$\frac{di_q}{dt} = \frac{1}{L_e} [v_q - Ri_q + \omega L_e i_d - \omega |\vec{\Psi}_r|]$$

Si reescribimos estas ecuaciones en el espacio de Laplace nos queda:

$$i_d \cdot (s \cdot L_e + R) = v_d + \omega L_e i_q$$

$$i_q \cdot (s \cdot L_e + R) = v_q + \omega L_e i_d - \omega |\vec{\Psi}_r|$$

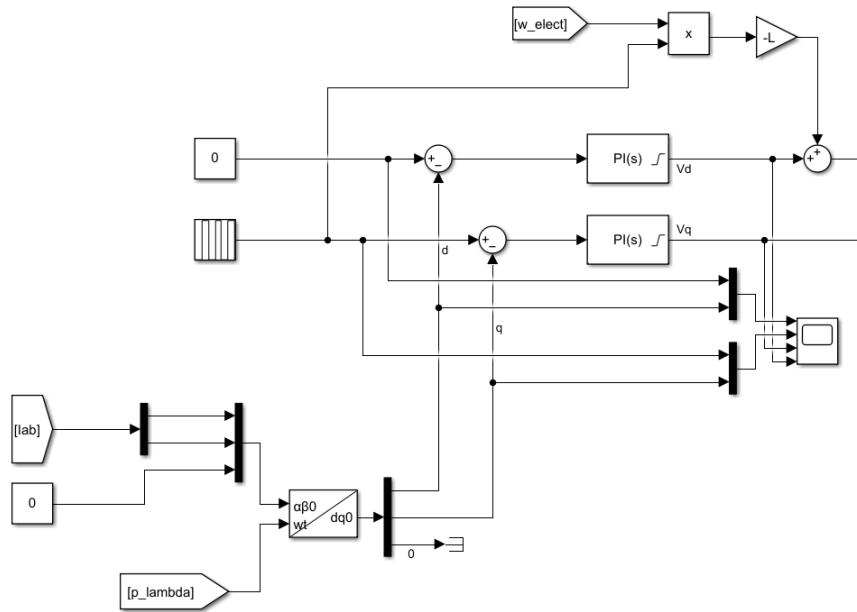
Recordemos que nuestra acción de control será la tensión. Cabe también remarcar que el comportamiento del sistema con relación a la corriente es mucho más rápido que el que va relacionado a la velocidad angular, esto quiere decir que podemos no tener en cuenta el último término de la última ecuación  $\omega |\vec{\Psi}_r|$  ya que su dinámica es mucho más lenta que la de los otros términos en lo que al control de corriente se refiere.

Además de esto hemos de tener en mente nuestro objetivo, que es controlar la corriente de eje Q que queramos y mantener la corriente de eje D nula. Esto implica que los términos cruzados en la parte del control de la corriente de eje Q relacionados con la corriente de eje D son despreciables, ya que, cuando funcione correctamente esta corriente deberá ser nula, esto nos deja con las siguientes ecuaciones simplificadas:

$$i_d \cdot (s \cdot L_e + R) = v_d + \omega L_e i_q$$

$$i_q \cdot (s \cdot L_e + R) = v_q$$

Podemos observar cómo, excepto por el término  $\omega L_e i_q$  de la primera ecuación nos queda un sistema de primer orden, cuyos parámetros además son constantes físicas del propio motor. Esto es ideal, ya que podemos obtener los parámetros del controlador a partir de mediciones realizadas empíricamente de forma sencilla. Para paliar el error que cometeríamos si no tuviéramos en cuenta el término  $\omega L_e i_q$  lo que haremos será compensarlo de forma externa al controlador ya que esto mejorará la respuesta dinámica del mismo. Visto en simulink quedaría tal que así:



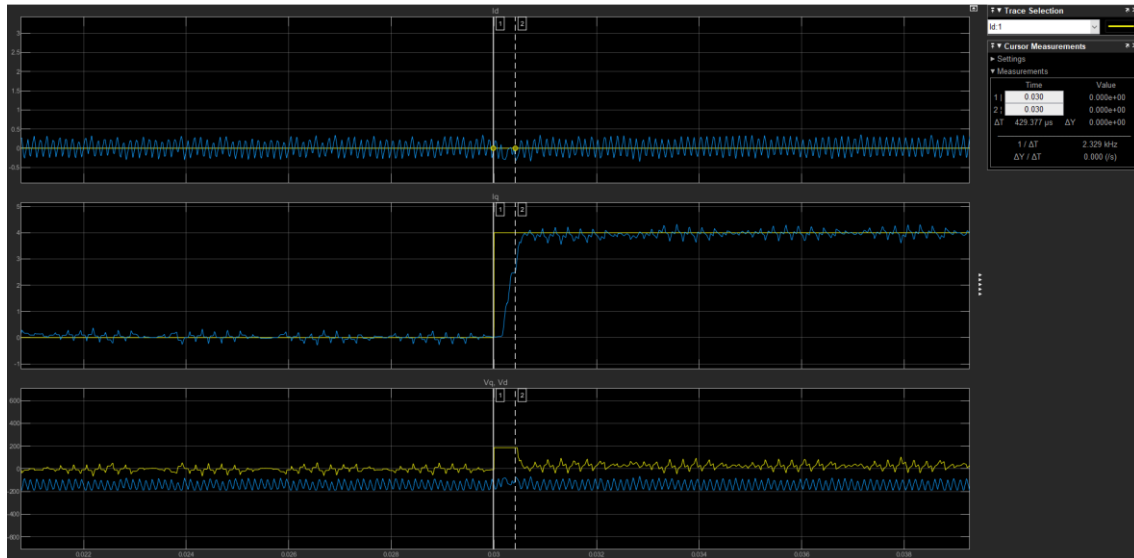
**Figura 63 Implementación del control por corriente en Simulink con representación de datos (elaboración propia)**

Hecho esto y recordando que los parámetros a introducir en los controladores PI tanto de eje Q como de eje D son: ganancia proporcional R y constante de tiempo L; podemos pasar a implementar también los límites de saturación de los controladores. Esto se realiza de forma muy sencilla en Matlab, sin embargo, a la hora de su implementación real deberemos de tener ciertos factores en cuenta.

Los límites que debemos introducir son dependientes del inversor que vayamos a utilizar, en nuestro caso al utilizar un inversor conectado a la red monofásica tenemos una tensión máxima de  $230V \cdot \sqrt{2} = 325V$  que da una tensión máxima de fase de  $\frac{325V}{\sqrt{3}} = 187V$ .

Luego este será nuestro límite de tensión que debemos comandar al inversor.

Veamos con estos datos y con una implementación sencilla gracias a librerías de Matlab del inversor cómo queda la respuesta de la corriente con nuestro controlador. Para ello generaremos una comanda de corriente nula en el eje D y una serie de pulsos cuadrados como comanda de corriente de eje Q.



**Figura 64 Captura de la simulación en Simulink (elaboración propia)**

En la imagen superior podemos distinguir 3 gráficos, el primero hace referencia a la corriente y eje D, en el que la señal de color amarillo es la referencia (que como se observa, es 0) y la azul es el valor real de la corriente en el eje. La segunda gráfica de la misma forma hace referencia a la corriente de eje Q, aquí sí que podemos observar el cambio en la referencia y la dinámica del sistema ante el cambio, como antes, la señal amarilla es la referencia y la azul el valor real. En la última gráfica podemos observar en amarillo la acción de control sobre la tensión en el eje Q y en azul observamos la acción de control sobre la tensión en eje D.

Podemos observar que el tiempo de respuesta del controlador es excepcional, en el orden de las centésimas de microsegundos. Además, es apreciable como ante cambios en la corriente demandada de 4 amperios el regulador trata de demandar la máxima tensión posible, sin embargo, esta se ve limitada a los 187V máximos que el inversor será capaz de entregar. Adicionalmente observamos cómo la corriente de eje D se mantiene constante entorno a los 0 amperios con el único rizado propio de la implementación del inversor y la generación del Space Vector PWM con el que éste trabaja. En la propia simulación vemos como, efectivamente, la respuesta que nos aporta es la propia de un sistema de primer orden, que frente a una comanda de escalón reacciona con una exponencial.

Con esto claro, podemos concluir que el control de corriente funciona y, por tanto, podemos pasar a la realización del control de velocidad del motor, que siempre fue el objetivo de este mismo trabajo.

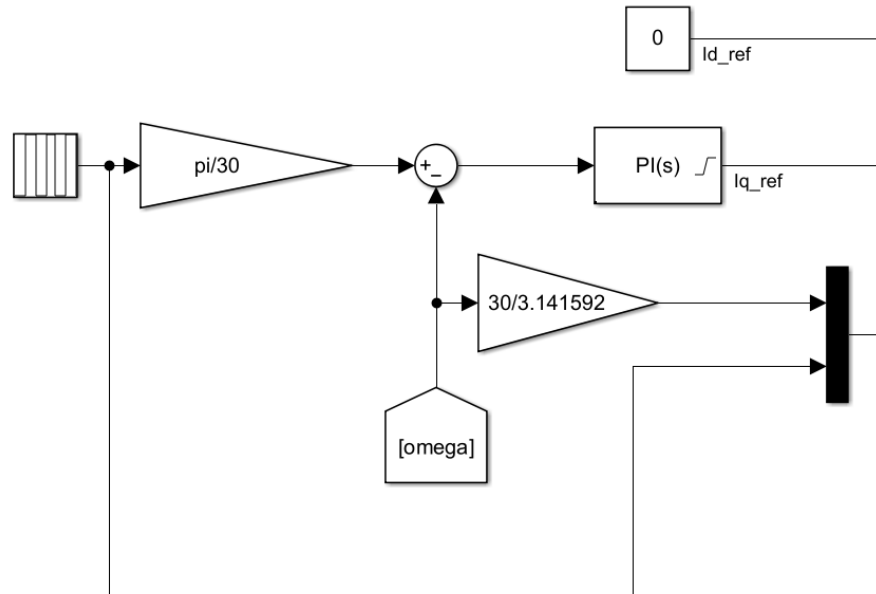
### **3.7. IMPLEMENTACIÓN EN SIMULINK DEL CONTROL DE VELOCIDAD**

El control de velocidad se basará una vez más en un controlador proporcional-integral, que esta vez tendrá como acción de control la corriente de eje Q, que recordemos es la que genera par motriz. Para ello, necesitamos realimentar la velocidad angular del eje del motor y que la acción de control del controlador de velocidad sea la referencia del controlador de corriente del eje Q.

Esta vez los parámetros del controlador no corresponden directamente a partes de la ecuación del sistema, debido a que tenemos retrasos secundarios en las distintas acciones de control,

medida, etc. Esto sin embargo nos permite estudiar cómo obtener los parámetros del controlador de forma empírica, a través de ensayos de prueba y error viendo cómo evoluciona la respuesta del sistema ante el escalón demandado.

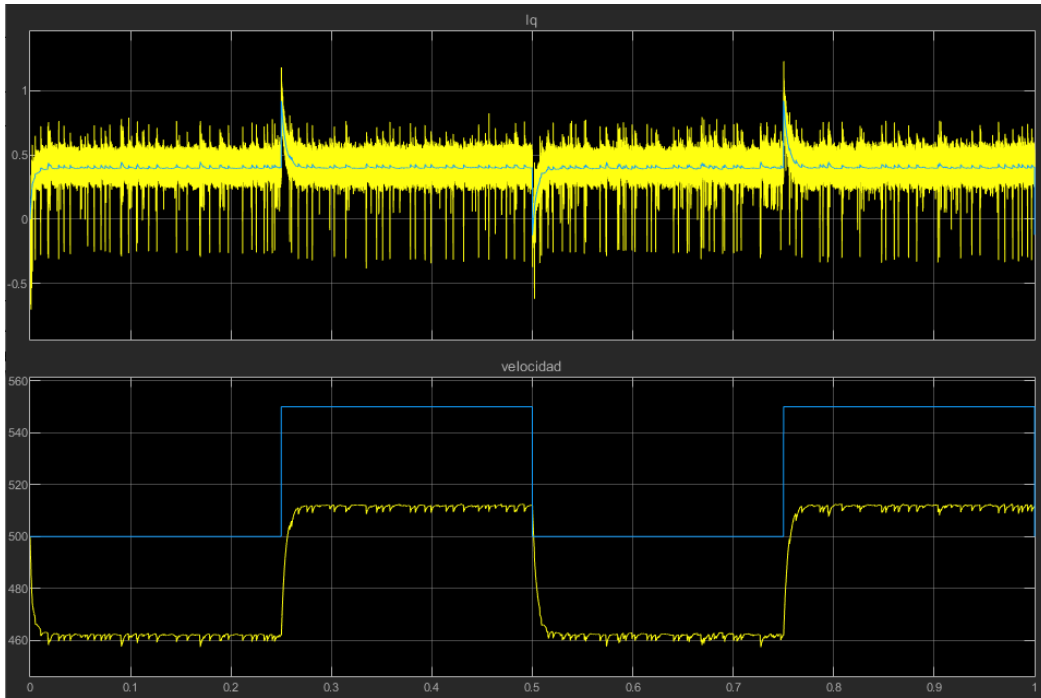
Volviendo al sistema de control, en Simulink debemos de mantener siempre constantes las unidades bajo las que trabajamos, en este caso utilizamos rad/s, sin embargo, habitualmente nos es más cómodo trabajar con revoluciones por minuto, esto nos lleva a realizar las transformaciones necesarias para obtener los datos en las unidades que buscamos, una vez hecho esto, obtenemos este bucle de control:



**Figura 65 Esquema del control de velocidad implementado en Simulink (elaboración propia)**

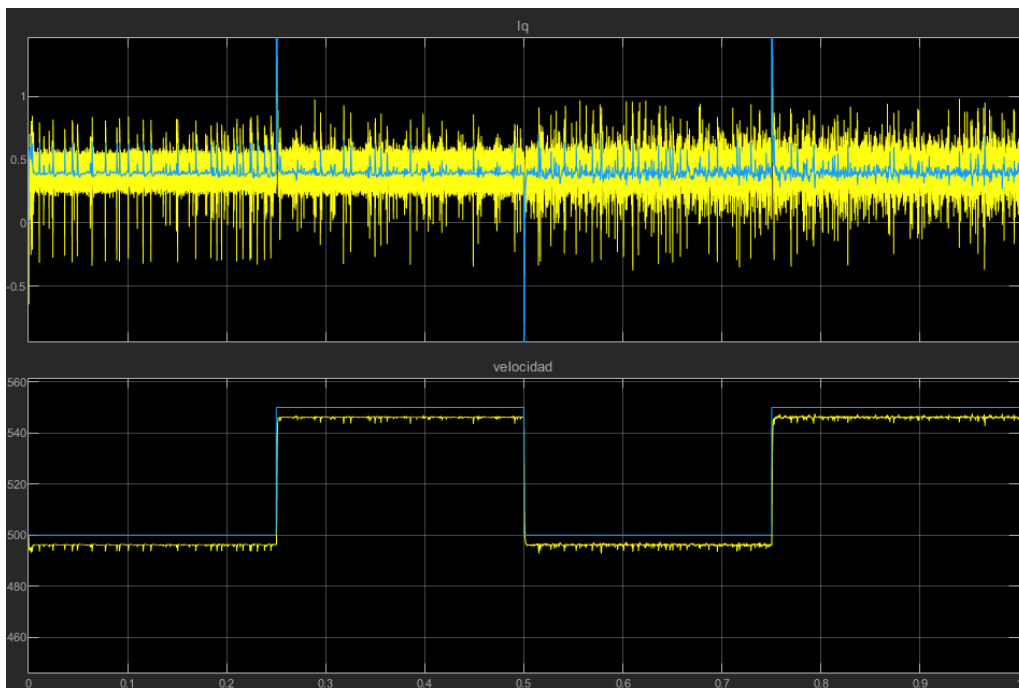
Una vez hecho esto, debemos proceder con la configuración del controlador. Como se ha mencionado previamente, esta se realizará de forma empírica. Para ser más fieles al funcionamiento real del equipo añadiremos un retraso de segundo orden en la realimentación de la velocidad, simulando los retrasos de cálculos del microcontrolador, lo haremos mediante un Zero-order hold. Comenzaremos únicamente con acción proporcional, dejando la constante de tiempo del controlador en un valor muy alto que no nos afecte en saltos rápidos a priori. Iremos aumentando la ganancia de nuestro controlador hasta que comencemos a percibir sobreoscilaciones. Una vez comencemos a encontrar sobreoscilaciones de entorno al 10/20% comenzaremos a añadir acción integral hasta que veamos que esta afecta negativamente a la respuesta del controlador. Comencemos por una ganancia relativamente baja de 0.1 para la parte proporcional y un tiempo integral muy grande, para que no tenga efecto a la hora de la respuesta del controlador:





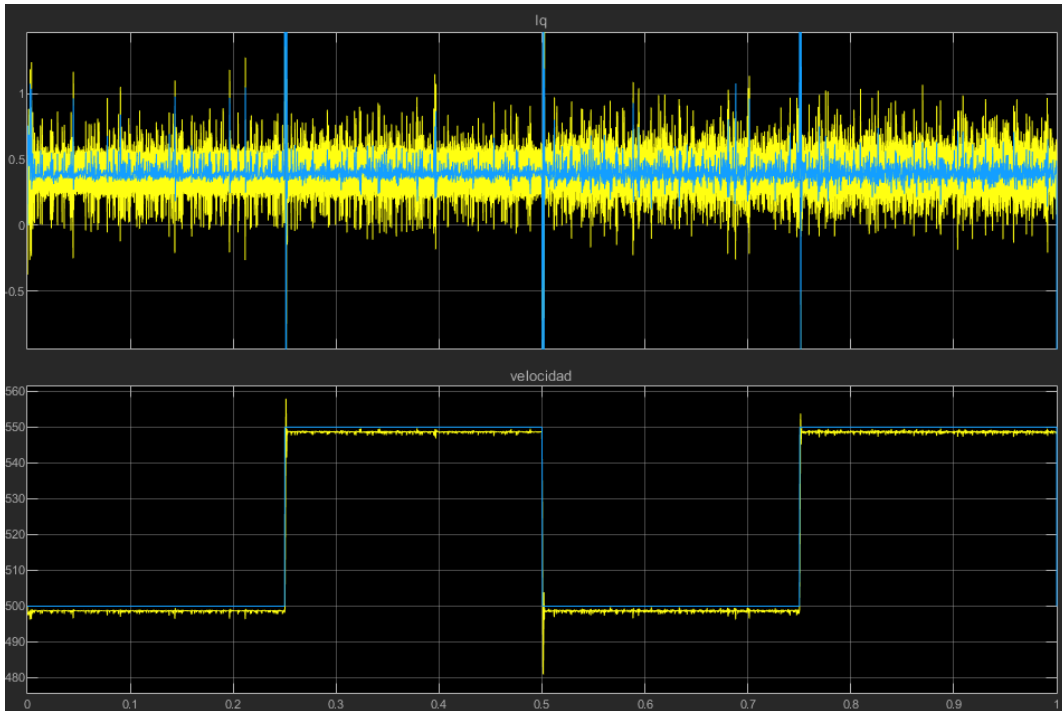
**Figura 66** Captura de la simulación del control de velocidad en Simulink (elaboración propia)

Se observa que, aunque efectivamente, no alcanzamos la velocidad objetivo exacta que se ha comandado (lógico, ya que no existe acción integral), no observamos tampoco una sobreoscilación en la velocidad. Aumentemos ahora la ganancia a 1.



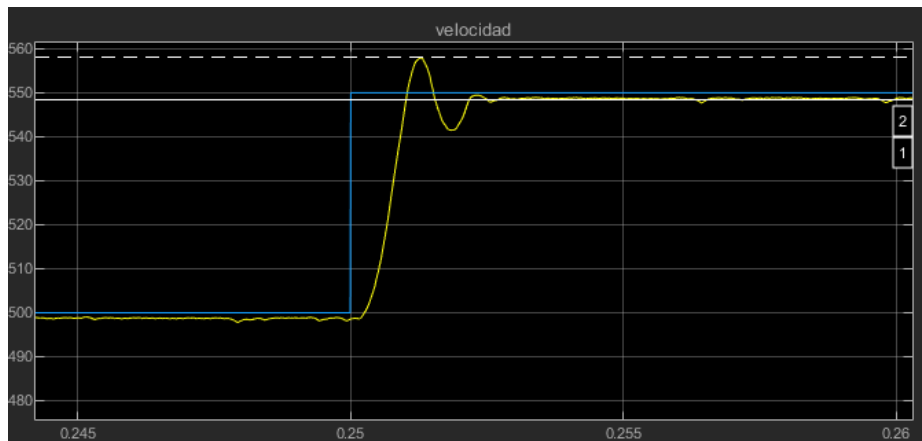
**Figura 67** Captura de la simulación del control de velocidad con segunda iteración de parámetros en Simulink (elaboración propia)

Observamos que, todavía no se detecta una sobreoscilación en la velocidad, aunque observamos que, al tener mayor velocidad de respuesta nos acercamos más a la velocidad objetivo. Sigamos subiendo la ganancia, ahora a 3.



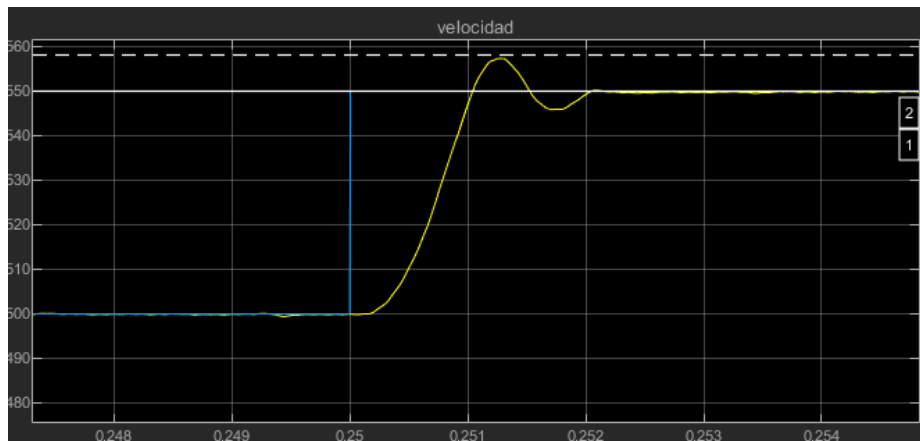
**Figura 68** Captura de la simulación del control de velocidad con tercera iteración de parámetros en Simulink (elaboración propia)

Ahora sí observamos una sobreoscilación en la velocidad, de hecho, si hacemos zoom en la parte del primer escalón podemos observar el porcentaje de sobreoscilación.



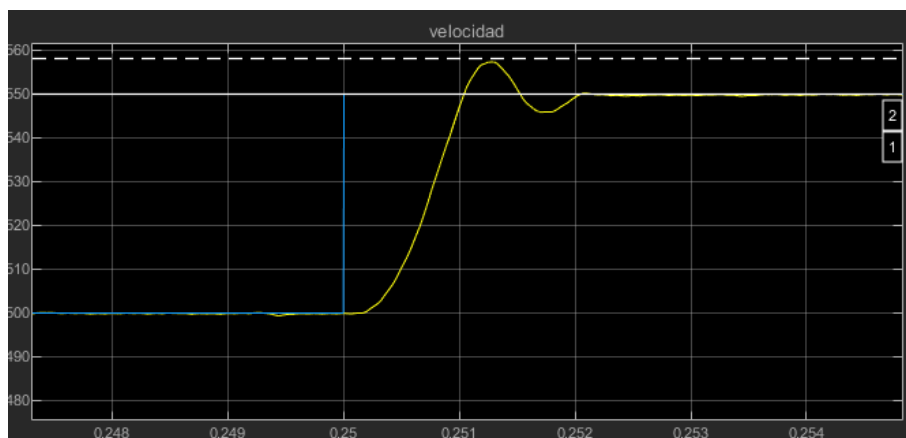
**Figura 69** Captura de la simulación de velocidad en Simulink con inciso en la sobreoscilación ante cambio en consigna (elaboración propia)

Se puede observar en las medidas de los cursores que la sobreoscilación es de unas 10 RPM, que sobre el salto de 50 RPM es un 20% de sobreoscilación. Ahora podemos continuar disminuyendo el tiempo característico hasta que observemos un aumento en la sobreoscilación. Ajustemos un tiempo de 0.1 s para nuestro tiempo de simulación.



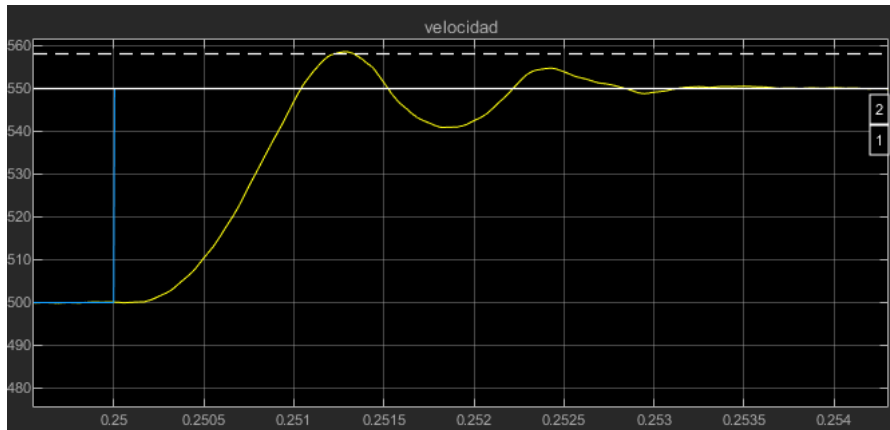
**Figura 70** Captura de la simulación de velocidad en Simulink (segunda iteración de parámetros del controlador) con inciso en la respuesta ante cambios de referencia (elaboración propia)

Como cabía esperar, para un tiempo de transición del orden de 0.05s no observamos todavía ningún cambio en la sobreoscilación significativa. Bajemos pues a 0.01s.



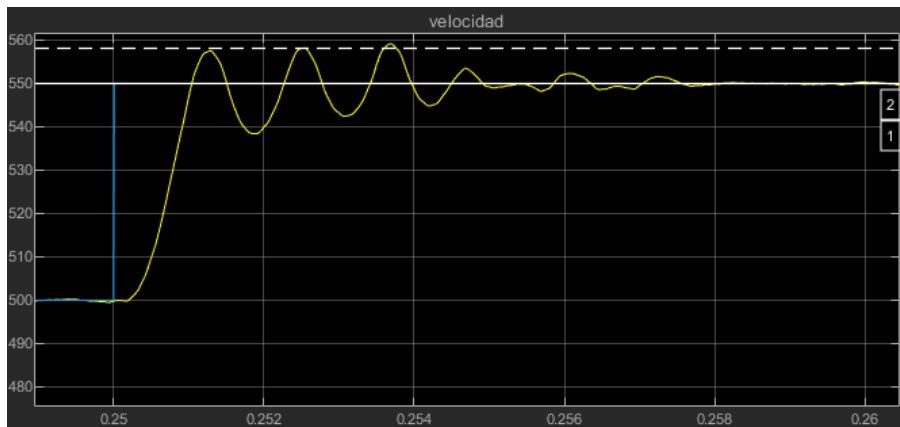
**Figura 71** Captura de la simulación de velocidad en Simulink (tercera iteración de parámetros del controlador) con inciso en la respuesta ante cambios de referencia (elaboración propia)

Como cabía esperar, para un tiempo de transición del orden de 0.05s no observamos todavía ningún cambio en la sobreoscilación significativa. Con un tiempo integral de 0.01s seguimos sin observar todavía ningún tipo de sobreoscilación mayor a la que observábamos anteriormente. Continuemos reduciendo el tiempo característico del controlador, esta vez a un valor más cercano al propio del sistema, unos 0.0005s y observemos cual es la reacción del sistema a esto.



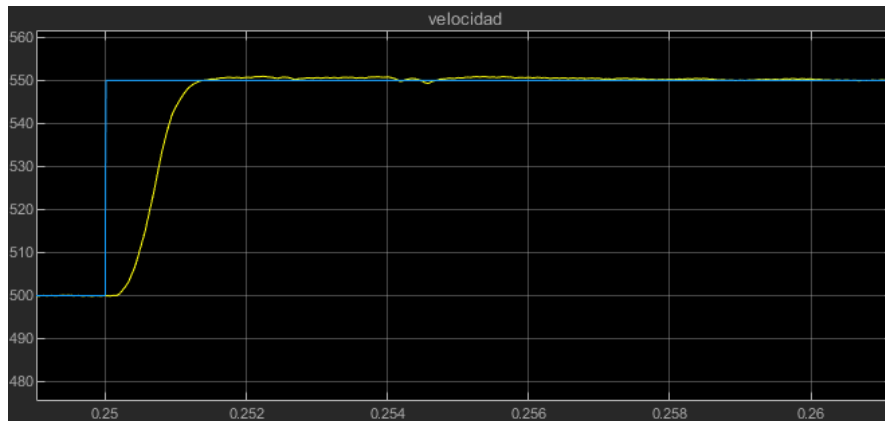
**Figura 72** Captura de la simulación de velocidad en Simulink (cuarta iteración de parámetros del controlador) con inciso en la respuesta ante cambios de referencia (elaboración propia)

Vemos que ahora sí se observa un aumento en la sobreoscilación. Podemos ver que la sobreoscilación tarda más en corregirse y que el módulo de la misma ha aumentado ligeramente. Para acabar de confirmarlo podemos probar a reducir el tiempo integral a 0.0003s y ver cuál es la respuesta.



**Figura 73** Captura de la simulación de velocidad en Simulink (quinta iteración de parámetros del controlador) con inciso en la respuesta ante cambios de referencia (elaboración propia)

Podemos observar cómo claramente se observa un descontrol en la velocidad, indicativo de que nos hemos excedido con el parámetro integral. En base a estos datos podemos escoger un compromiso entre los datos que hemos ido recogiendo. Si asignamos una ganancia de 1.5 y un tiempo integral de 0.005s podemos observar una respuesta ante cambios de velocidad excelente:



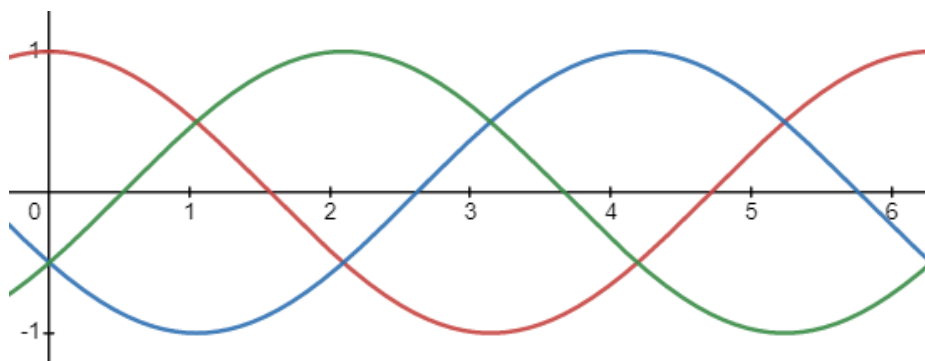
**Figura 74 Captura de la simulación de velocidad en Simulink con el controlador ajustado (elaboración propia)**

Con esto aclarado podemos dar la fase de simulación por concluida.

### 3.8. IMPLEMENTACIÓN TEÓRICA DE LA MODULACIÓN DE VECTOR ESPACIAL

La modulación de vector espacial es una técnica de operación de inversores que facilita el control del campo magnético y del par en motores como el que nos ocupa, del estilo PMSM, y también para muchas otras tipologías de máquinas eléctricas como las de inducción. Esta técnica se utiliza para controlar las conmutaciones de los interruptores del inversor para generar la forma de onda que buscamos.

Lo más natural es pensar en generar una serie de tensiones trifásicas con forma senoidal separadas  $120^\circ$ , lo que sería un sistema trifásico convencional, lo que se podría conseguir con una modulación PWM por portadora triangular monofásica en cada una de las tres ramas del inversor:



**Figura 75 Ejemplo de función de onda trifásica senoidal (elaboración propia)**

El mayor problema de esta forma de generar las tensiones trifásicas es que no llegas a utilizar el 100% de la tensión disponible en el bus de continua en ningún momento si lo que se desea es aplicar un sistema senoidal equilibrado. Esto es porque la tensión realmente se aplica entre dos puntos, en este caso entre dos de las tres fases, haciendo uso de esta forma de generar corrientes trifásicas la máxima diferencia de tensión entre fases es únicamente del 86.6% del total que disponemos. Podemos visualizar este extremo con un ejemplo numérico: consideremos una red trifásica de 400V: para esa tensión de línea, la tensión en el bus de

continua es la que proporciona la etapa de entrada típica, un rectificador en puente no controlado y es igual a máximo de la tensión de línea:  $400 \cdot \sqrt{2} = 565V$ . Para obtener la tensión en cada terminal de las tres fases se puede utilizar, como se ha indicado, la modulación por portadora triangular y, en el caso de la mayor moduladora aceptable (de amplitud igual a la portadora) se tendría una tensión de fase, respecto del punto medio del bus de continua, de  $565V/2 = 282,5V$ . Ahora bien, este sería el valor máximo de la tensión de fase, que, para una variación senoidal con el tiempo, le corresponde un valor eficaz de  $282,5/\sqrt{2} = 200V$ , es decir, un 15% del valor *natural* de la tensión simple en un sistema de 400V de tensión de línea, 230V. Esto ocurre, intuitivamente, debido a que las partes máximas de las ondas alcanzan valores altos sin llegar a generar realmente una diferencia de tensión notable. Para evitarlo es razonable pensar que debemos tratar de minimizar estas cumbres de las ondas. Para hacer esto debemos de realizar algún tipo de operación a cada una de las formas de onda que no afecte a la forma de onda de la tensión entre fase y fase.

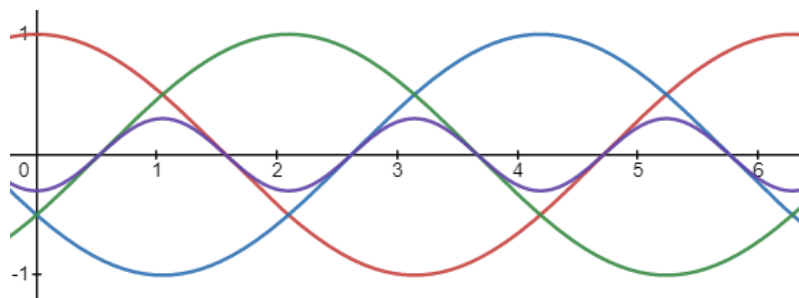
Una de las formas de realizar esto es mediante la suma de una forma de onda de tercer armónico, así las tres fases comparten la misma suma (debido a que en un sistema trifásico los terceros armónicos pulsan en fase en las tres fases) y al realizar la diferencia para obtener la tensión entre ellas esta suma se cancela. Nuestras tres tensiones de fases iniciales serían:

$$\begin{aligned} V_R &= A \cdot \cos(\omega t + 0) \\ V_S &= A \cdot \cos(\omega t + 120^\circ) \\ V_T &= A \cdot \cos(\omega t + 240^\circ) \end{aligned}$$

Si añadimos esta forma de onda que llamaremos  $V_o$  a las tres fases tenemos:

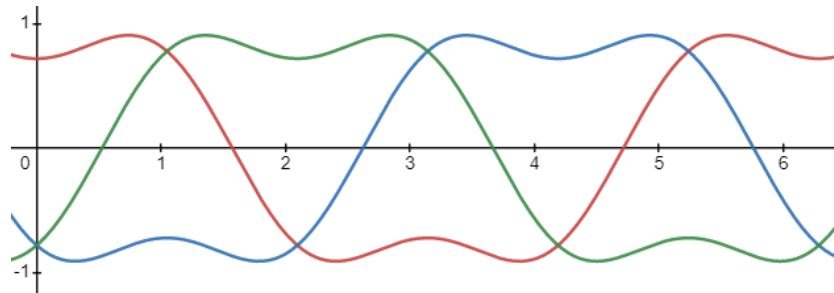
$$\begin{aligned} V_R &= A \cdot \cos(\omega t + 0) + V_o \\ V_S &= A \cdot \cos(\omega t + 120^\circ) + V_o \\ V_T &= A \cdot \cos(\omega t + 240^\circ) + V_o \end{aligned}$$

Veamos cómo queda esta onda  $V_o$  de forma gráfica comparada con las ondas senoidales:



**Figura 76 Comparación entre  $V_o$  y las ondas trifásicas senoidales (elaboración propia)**

Veamos entonces cómo queda la suma de esta onda con las demás:



**Figura 77 Suma de las formas de ondas trifásicas senoidales y  $V_0$  (elaboración propia)**

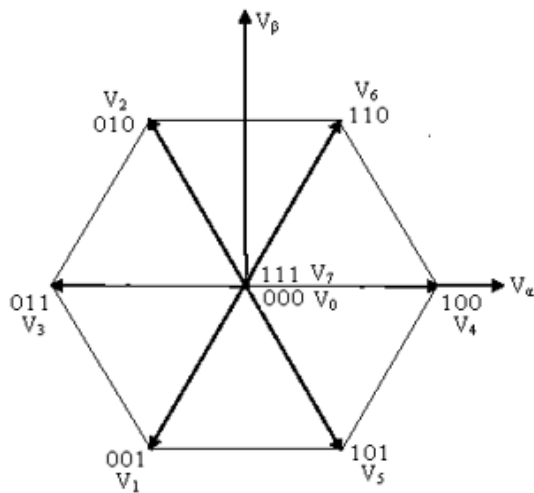
Aunque no de forma precisa, ya que los valores son arbitrarios, podemos observar cómo los valores máximos de las ondas han disminuido drásticamente, y sin embargo vemos lo que ocurre con las diferencias entre una tensión de fase y otra:

$$V_{RS} = A \cdot \cos(\omega t + 0) + V_{\phi} - A \cdot \cos(\omega t + 120^\circ) - V_{\phi}$$

Se observa que a la diferencia de tensiones entre fases no le ha afectado la suma que hemos realizado de este tercer armónico.

Con esto claro, debemos entender que lo que buscamos nosotros es generar los pulsos que, aplicados a los interruptores del inversor y una vez filtrada la tensión que generemos, nos darán la forma de onda que acabamos de observar.

Para esto hay que tener claro que posibilidades de conmutación tenemos:



Conmutación	R	S	T
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

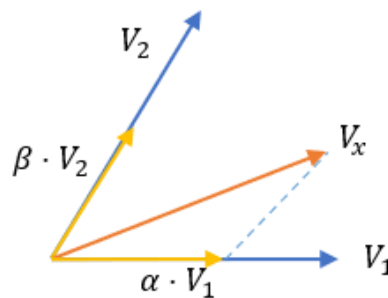
**Figura 78 Secuencia de conmutaciones de las ramas del inversor (elaboración propia)**

Estas son las posibles conmutaciones que tenemos para los diferentes interruptores del inversor. Como vemos son 8 posibilidades, de las cuales 2 tienen el mismo efecto, la 0 y la 7. Esto es porque ambas tienen el efecto de no crear ninguna diferencia de tensión entre fases. Cabe recordar que lo que buscamos al final es crear un vector rotativo que gire dentro del hexágono que hemos generado con los vectores anteriores. Este vector giratorio lo podemos generar a través de componer (repartir en el tiempo correspondiente a un periodo de modulación) los dos vectores contiguos más cercanos al vector en el punto en el que lo queremos generar. Además, la forma hexagonal que hemos definido no ha sido de forma arbitraria, si nos fijamos la forma hexagonal es el límite del recorrido máximo del vector giratorio que podemos generar. Esto es así debido a que como se ha mencionado antes, para generar un vector intermedio entre dos de las 8 conmutaciones disponibles debemos de generarlo mediante la modulación de los dos contiguos y como sabemos, la modulación del vector, por ejemplo  $V_x$  se realiza de esta manera:

$$V_x = \alpha \cdot V_1 + \beta \cdot V_2$$

Siendo siempre la suma de  $\alpha + \beta$  inferior o igual a la unidad, ya que recordemos que hacen referencia al porcentaje de tiempo o "duty cycle" que están activos cada uno y, entre los dos, no pueden estar activos más del 100% del periodo de modulación.

Si lo queremos observar de forma gráfica, lo que buscamos es esto, siendo el ángulo entre  $V_1$  y  $V_2$  de 60 grados:



**Figura 79 Esquema vectorial del vector de tensión en el primer sextante (elaboración propia)**

En el caso de que no queramos un vector del tamaño máximo permitido físicamente, es decir, un vector que no esté ceñido estrictamente al borde del hexágono debemos de añadir otro término más a nuestra ecuación, el término nulo. Como hemos comentado antes constamos de dos términos nulos  $V_0$  y  $V_7$ . La ecuación quedaría de la siguiente manera:

$$V_x = \alpha \cdot V_1 + \beta \cdot V_2 + \gamma \cdot V_{1|7}$$

Para cada sextante despejaremos a partir de las coordenadas cartesianas los valores de  $\alpha + \beta + \gamma$ . Con esto claro podemos pasar a generar la modulación por vector espacial.

Para generar la modulación de vector espacial por ancho de pulso debemos de, a través de lo anteriormente expuesto, generar las conmutaciones de cada una de las tres fases. Para ello lo que haremos será, en cada uno de los sextantes, recorrerlo de forma que solo uno de los interruptores cambie a la vez. De esta forma generaremos una secuencia que recorrerá el sextante de forma triangular empezando y acabando en el mismo punto, que siempre será el vector 1 o 7. Lo podemos observar con el ejemplo del primer cuadrante. La secuencia a recorrer sería:  $V_0 - V_4 - V_6 - V_7$ . Como vemos, comenzamos en el mismo punto del hexágono que en el



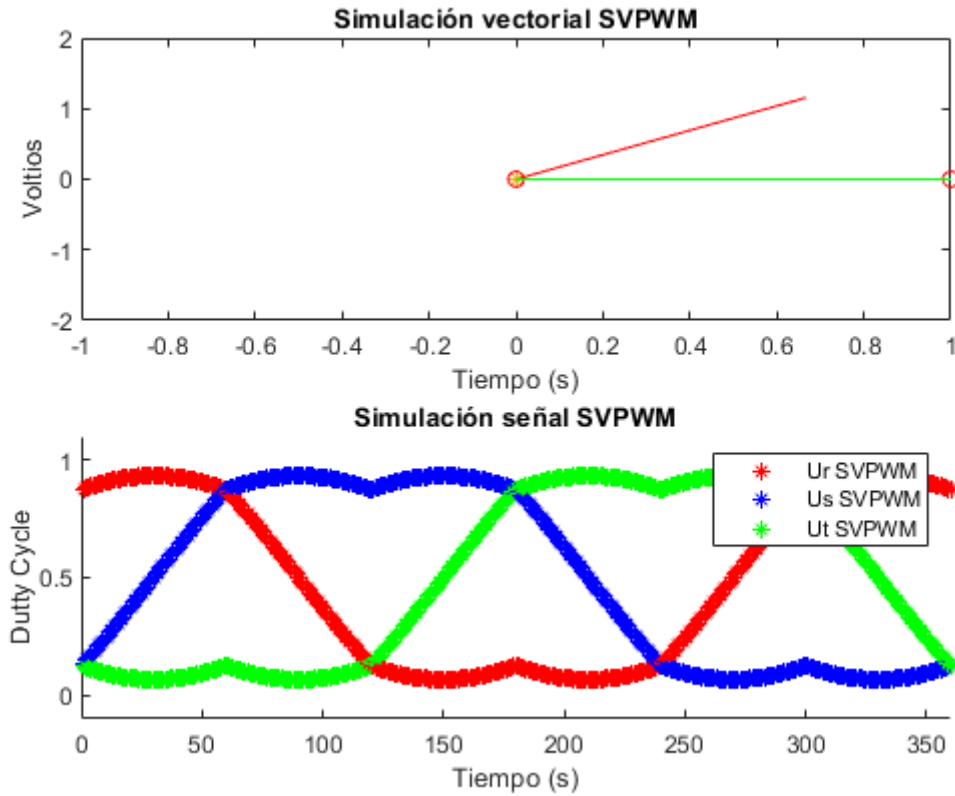
que acabamos, aunque no sea el mismo en cuanto a combinaciones de interruptores. Recordemos también, que como se ha mencionado antes en la combinación vectorial  $\alpha$  hace referencia al primer vector,  $\beta$  al segundo y  $\gamma$  a la parte nula. En este caso como constamos de dos partes que actúan como parte nula, lo razonable será dividir en partes iguales el tiempo que queremos que esté cada una activa, siendo  $\frac{\gamma}{2}$  el correspondiente a cada una. Y si lo expresamos en forma de tabla podemos ver la correlación que buscábamos

VECTOR	$\alpha, \beta, \gamma$	R	S	T
$V_0$	$\frac{\gamma}{2}$	0	0	0
$V_4$	$\alpha$	1	0	0
$V_6$	$\beta$	1	1	0
$V_7$	$\frac{\gamma}{2}$	1	1	1

**Tabla 1 Tiempos de conmutación de las ramas del inversor para el primer cuadrante**

Las columnas R, S y T indican los estados de los semiconductores de potencia (interruptores cuando trabajan en modo conmutación) de la rama del inversor que deben de conmutar y el tiempo de conmutación viene dado por  $\alpha, \beta$  y  $\gamma$ . De esta forma, podemos observar que para el ejemplo del primer sextante el tiempo de conmutación para los interruptores de la rama de la fase R es  $\alpha + \beta + \frac{\gamma}{2}$  activo y  $\frac{\gamma}{2}$  desactivado. Se habla de interruptores en plural ya que cada rama está formada por 2 interruptores, los cuales jamás estarán conectados simultáneamente y que cuando cerremos uno de ellos el otro debe de abrir para evitar un cortocircuito en el bus de continua.

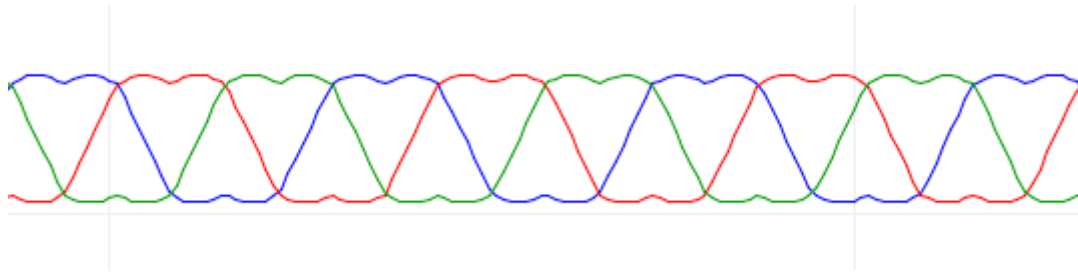
Para comprobar que lo que se acaba de explicar realmente funciona podemos generar con Matlab un programa que reproduzca lo que haremos eventualmente con nuestro microcontrolador. Generaremos un bucle en el que tendremos un vector que irá girando hasta completar 360 grados. En el bucle se comprobará el sextante en el que se encuentra el vector giratorio, una vez se comprueba se calculan Alpha, beta y gamma. Una vez hecho esto ya podemos calcular los tiempos de conmutación de las tres fases. Si graficamos los tiempos de conmutación debemos observar la forma de onda con jorobas en sus puntos máximos y mínimos cuyo valor oscile entre valores cercanos a 0 para el mínimo y cercanos a 1 para el máximo.



**Figura 80 Simulación en Matlab del control SVPWM (elaboración propia)**

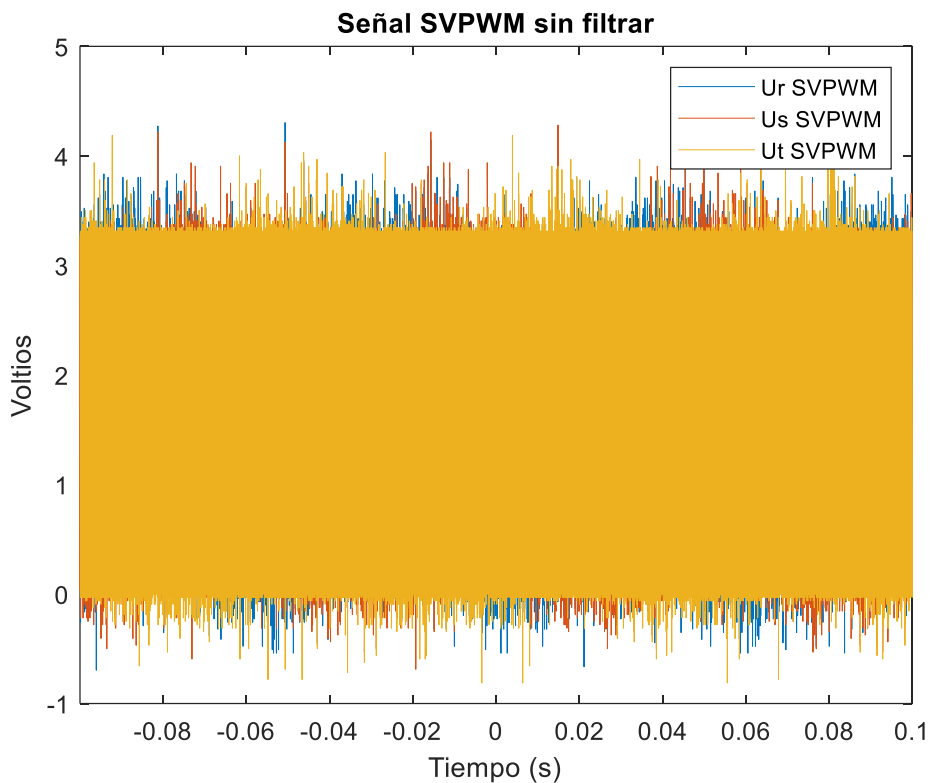
En el gráfico de arriba se generan 3 vectores, dos de ellos que delimitan el sextante en el que se encuentra el vector y el propio vector. En el gráfico 2 de ellos coinciden. En el gráfico inferior se puede observar que, efectivamente observamos la forma de onda que esperábamos. Vemos las dos jorobas que se comentaron previamente y que su valor oscila entre casi 0 y casi 1.

Al haber comprobado que funciona lo sensato ahora sería comprobar que, ya que tenemos la realimentación de posición del motor funcionando, podemos generar esto mismo a través del microcontrolador. Para ello lo que hacemos es, antes que nada, calibrar el sensor de posición que recordemos viene dado por el servopack. Para calibrarlo debemos de dar una vuelta completa para que el servopack nos envíe la señal de paso por 0, a partir de ese momento el microcontrolador ya sabe en qué posición se encuentra el motor. Una vez conocida la posición concreta generamos una función que, a través del ángulo nos indique en cuál de los 6 sextantes se encuentra el vector de posición del eje del motor. Una vez conocido el sextante nos crearemos un case-switch en el que, en función del sextante en el que nos encontremos se realice el mismo cálculo que hemos realizado en Matlab. Nos generará 3 valores entre 0 y 1 que podemos graficar mediante la función Serial Plotter del visualizador COM de Arduino.



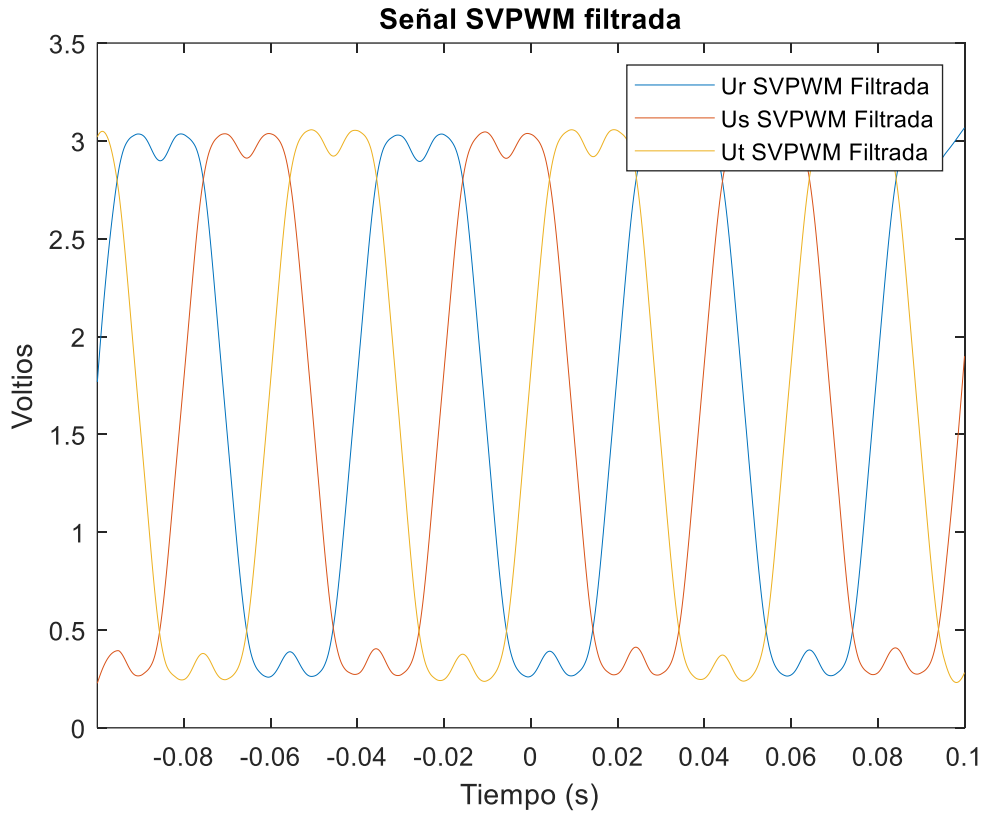
**Figura 81** Captura de la función Serial Plotter de la aplicación Arduino IDE de los tiempos de activación de las distintas ramas del inversor (elaboración propia)

Se puede intuir que las formas de onda son correctas. Lo que podemos hacer a continuación es medir también las salidas que el microcontrolador envía a las entradas de los interruptores del inversor. Esta señal no se verá claramente en el osciloscopio ya que lo único que observaremos será la señal de conmutación con una componente muy intensa a la frecuencia de conmutación que enmascara la componente de baja frecuencia que contiene en cada fase:



**Figura 82** Registro de señales SVPWM generadas por el ESP32 sin filtrar (elaboración propia)

Sin embargo, lo que podemos hacer es filtrar a través de la transformada de Fourier las frecuencias más altas de la señal, dejando únicamente la forma de onda que buscábamos. Para esto eliminaremos los términos de orden superior al 5, quedándonos esta forma de onda tras filtrar:

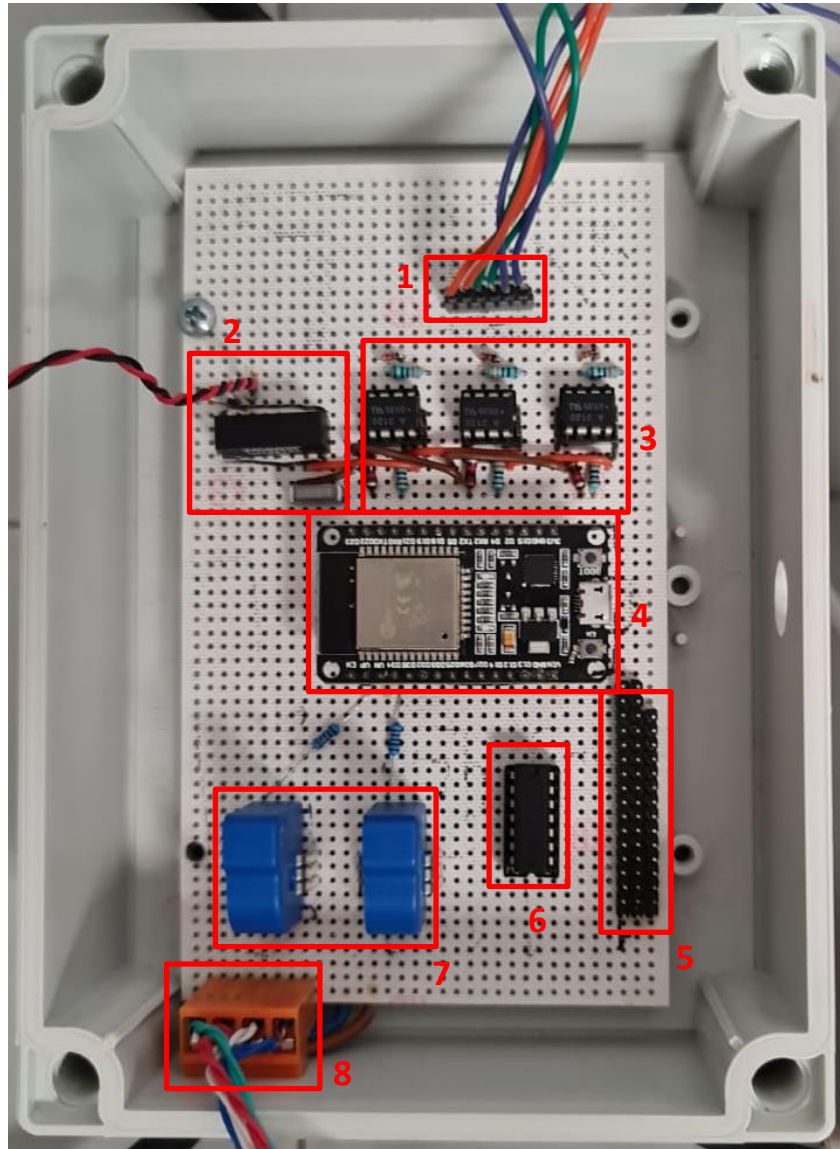


**Figura 83 Registro de señales SVPWM generadas por el ESP32 filtradas (elaboración propia)**

Observado esto podemos concluir que realmente sí estamos generando de forma adecuada las formas de onda necesarias para el inversor (que por supuesto no conectaremos hasta tener todas las precauciones controladas).

### 3.9. DESCRIPCIÓN DEL PROTOTIPO

El prototipo montado consta de 8 partes principales, entre las que se encuentran los conectores (marcados como 1, 5 y 8), los optoacopladores (marcados como 3), alimentación (marcada como 2), la placa de evaluación del microcontrolador ESP32 (marcada como 4), las puertas lógicas (marcadas como 6) y los sensores de corriente de efecto hall (marcados como 7).



**Figura 84 Foto de la tarjeta con los componentes soldados y numerados (elaboración propia)**

Comencemos comentando los conectores que integra la tarjeta. El conector marcado como 1 es el encargado de recibir las señales de pulsos de posición del encoder, estas señales como se comentó en el capítulo 1 vienen de forma diferencial, y se puede apreciar en los colores de los cables, contamos con 6 cables combinados por pares de color iguales que transmiten las 3 señales A, B y paso por 0. El conector 5 es el encargado de transmitir los datos generados por el microcontrolador al inversor a través de un cable plano (no mostrado en la imagen). Y por último en cuanto a conectores, el marcado como 8 es el encargado de recibir los cables de potencia del motor para hacer pasar la corriente de 2 de ellos a través de los sensores de corriente e interconectar la fase restante y la señal de tierra al cable de potencia soldado bajo la placa.

Marcado como 2 tenemos un DC-DC boost que recibe la alimentación de 5V DC de un USB externo y la transforma a 24V DC para la alimentación de los optoacopladores. Este DC-DC tipo boost es un TMR 6-0515 de la marca TRACO POWER, este es un convertidor aislado que permite separar la tensión de 24V de la de 5V completamente. Adicionalmente, tenemos marcado como 3 los tres optoacopladores digitales HCPL-3120 para el aislamiento de las señales de posición provenientes del encoder. Y por último tenemos las puertas lógicas OR marcadas en la figura como 6, que en este montaje no se utilizan, pero fueron montadas para maximizar la compatibilidad con otras tecnologías de control, en concreto para el control de máquinas de corriente continua sin escobillas o Brushless DC (BLDC).

El esquema de la tarjeta está adjunto en el apartado de planos.

### 3.10. SENSOR DE CORRIENTE

Para poder continuar con la implementación del controlador lo último que nos queda es la capacidad de lectura de corriente. Esto es vital por supuesto para la realimentación del controlador. Para realizar la medición de corriente utilizaremos un sensor de corriente LEM LTS 6-NP, este sensor requiere de la intersección del cable a través del cual circula la corriente. Tiene 6 puntos de conexión que permiten dar hasta 3 vueltas por el primario, de forma que se pueda ajustar la corriente de forma adecuada al rango de medida del sensor. El sensor en cuestión es capaz de leer corrientes tanto positivas como negativas, para hacerlo el datasheet del componente nos indica que parte de una tensión media de 2.5V, la cual va aumentando y disminuyendo en función de la corriente que circule a través del primario del sensor:

#### Output voltage - Primary current

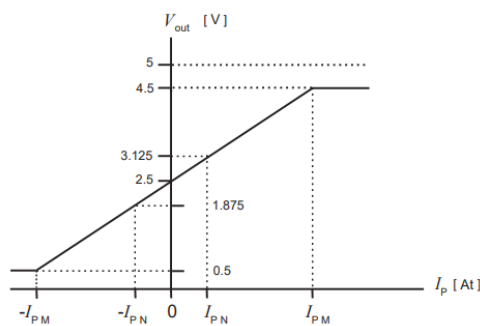


Figura 85 Gráfica de linealidad tensión-Corriente (LEM)

Esto es un problema para nuestro montaje, ya que el microcontrolador que utilizamos tiene entradas analógicas de un máximo de 3.3V. Para ajustar la tensión lo que haremos será generar un divisor de tensión de forma que, esos 2.5V pasen a ser 1.65V, que es el punto medio de los 3.3V máximos del rango de entrada de las entradas analógicas. El esquemático del sensor por lo tanto nos quedará tal que así:

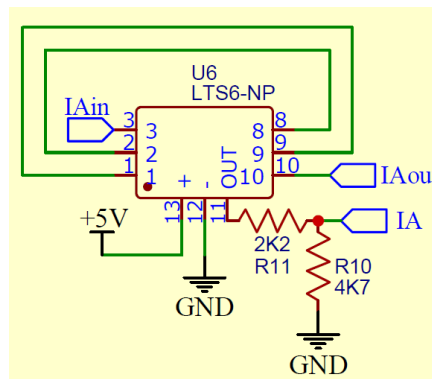


Figura 86 Esquema de conexión del sensor de corriente (elaboración propia)

El cálculo del divisor de tensión es el siguiente:

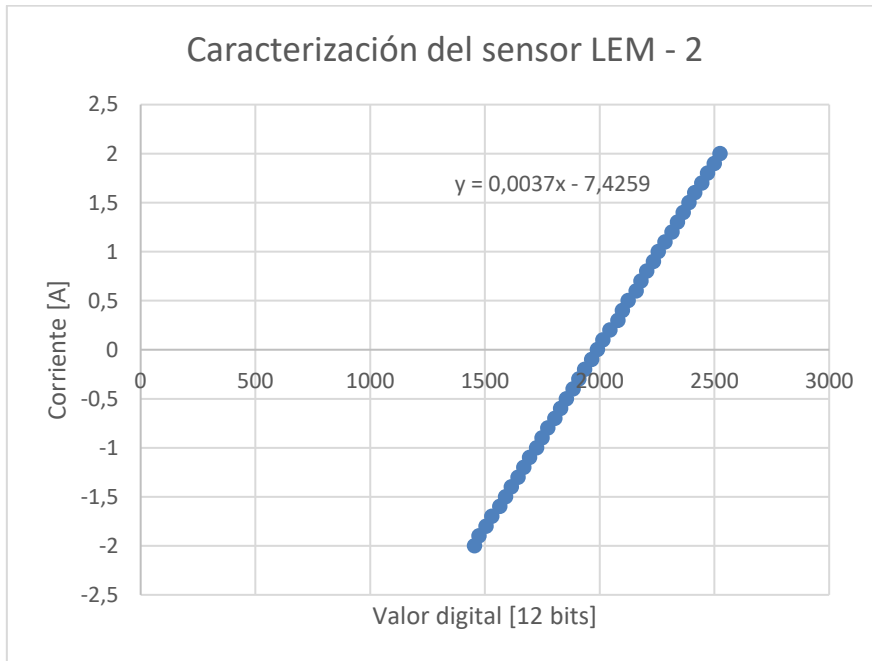
$$V_{out} = \frac{4700}{2200 + 4700} \cdot V_{in}$$

Realmente el cálculo que se ha mencionado anteriormente a cerca de los 1.65V sería el preciso que se requeriría, sin embargo, nos debemos ajustar a las resistencias típicas de la serie E12 que tenemos en el laboratorio.

Con el circuito claro y montado, lo que haremos será caracterizar el sensor, es decir. Con ayuda de la fuente de alimentación con limitación de corriente y un multímetro con capacidad de lectura de corriente iremos tomando medidas de la corriente y de los valores digitales que obtenemos de la lectura analógica a través del microcontrolador.

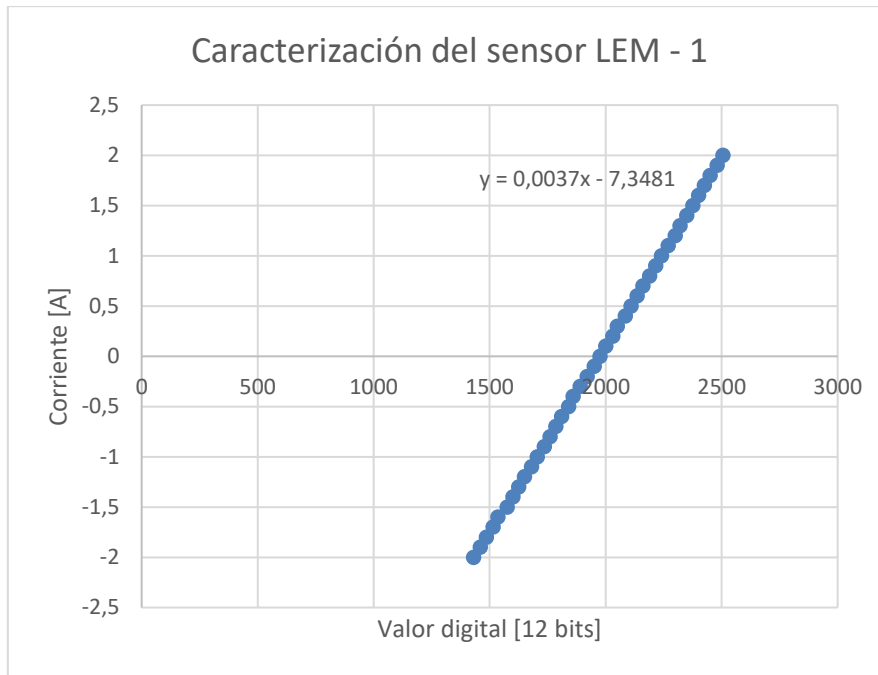
Para aprovechar el ensayo lo que haremos será realizar una conexión de forma que la corriente que “entra” por uno de los sensores “sale” por el otro. Así conseguimos que un mismo ensayo de corriente nos dé información sobre corrientes en ambos sensores y tanto en positivo como en negativo. Debemos esperar, según el datasheet del fabricante una respuesta prácticamente lineal en tensión con respecto a la corriente.

Los datos del ensayo son los siguientes:



**Figura 87 Caracterización del sensor LEM 2 (elaboración propia)**

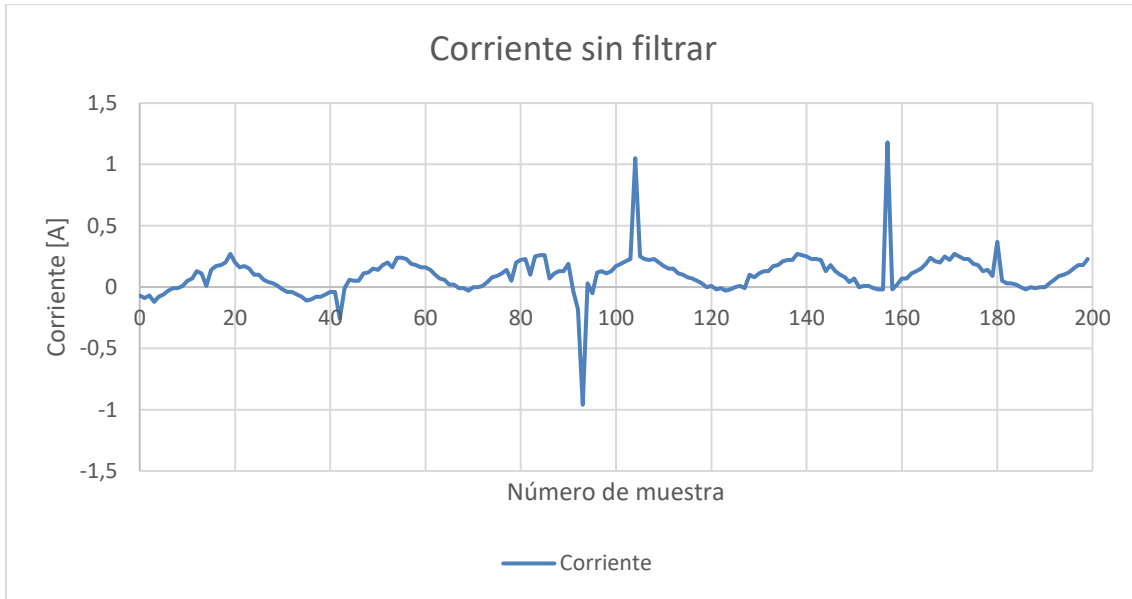




**Figura 88 Caracterización del sensor LEM 1 (elaboración propia)**

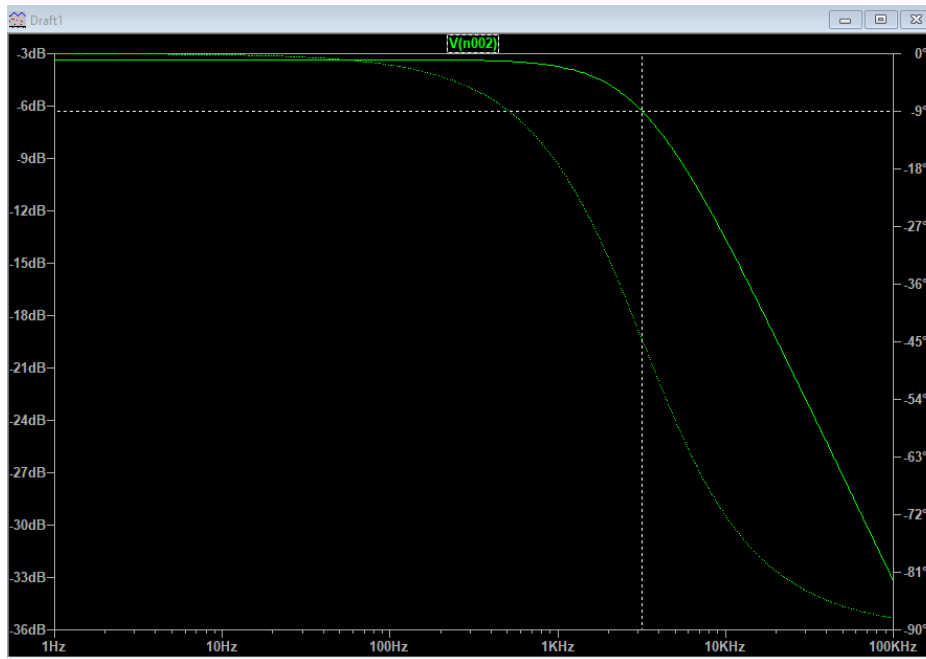
Observamos una clara linealidad entre la tensión y la corriente, lo cual es perfecto para nuestro propósito. Con estos datos obtenemos la fórmula de la línea de tendencia característica de cada sensor. Cabe remarcar que estos ensayos se llevaron a cabo con la alimentación proporcionada por el cable USB y no por el inversor. Esto fue realmente un fallo ya que la alimentación por USB al microcontrolador pasa a través de diodos de protección de tensión inversa que causan una caída de tensión. Esta caída de tensión no es importante para el microcontrolador, ya que éste consta de una alimentación regulada a 3.3V, sin embargo, los sensores sí trabajan a 5V, y cualquier desviación en la alimentación causará una desviación en su respuesta a la corriente. Esto hará que cuando en lugar de alimentar mediante el cable USB los sensores lo hagamos a través de los 5V que el inversor proporciona los valores digitales correspondientes a 0A puedan estar ligeramente modificados y que sea necesario realizar algún tipo de calibración extra.

Una vez calibrado debidamente se realizaron pruebas posteriores con saltos de corriente para observar a respuesta del sensor ante corrientes proporcionadas por el inversor.



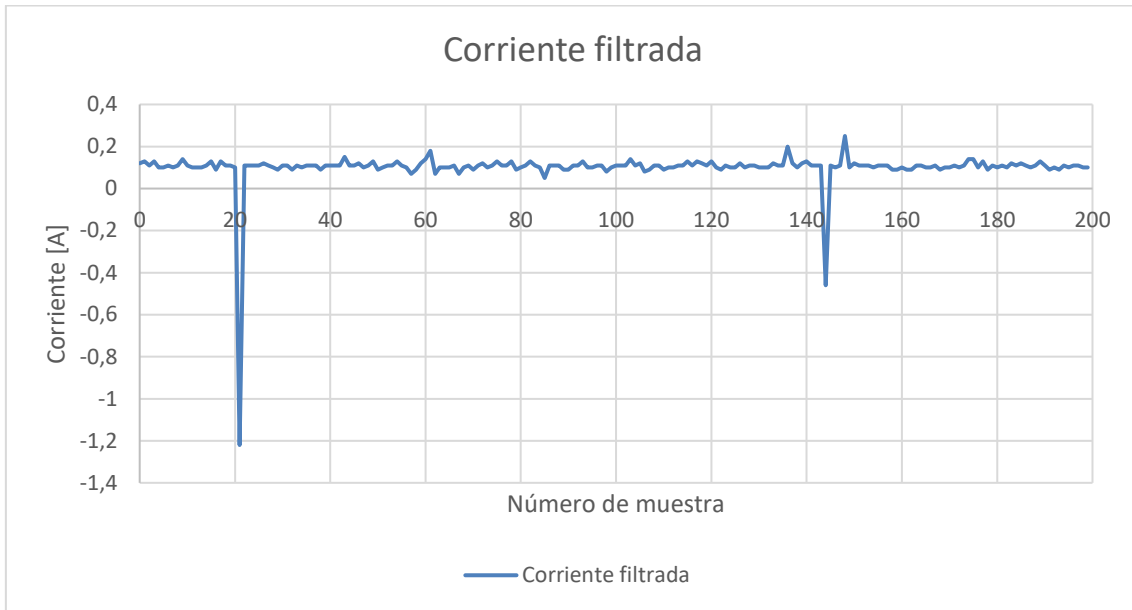
**Figura 89 Gráfica de corriente leída por el sensor LEM sin filtro (elaboración propia)**

Se puede observar que hay tanto ruido como un componente ondulatorio de error considerable (éste último debido probablemente al Aliasing que se produce al muestrear la señal de corriente con una frecuencia de rizado debida a la conmutación que empieza a 10 kHz utilizando una frecuencia de muestreo, 5kHz, muy inferior a la frecuencia del rizado). Para eliminar este ruido y este componente lo que haremos será implementar un filtro paso bajo además del divisor de tensión implementado anteriormente. Para implementar el filtro paso bajo pondremos en paralelo un condensador de 33pF con la resistencia de 4k7 ohmios, esto según las simulaciones que se realizaron nos debe dar una frecuencia de corte de 3kHz, suficiente como para eliminar los artefactos (Aliasing) posiblemente creados por el inversor:



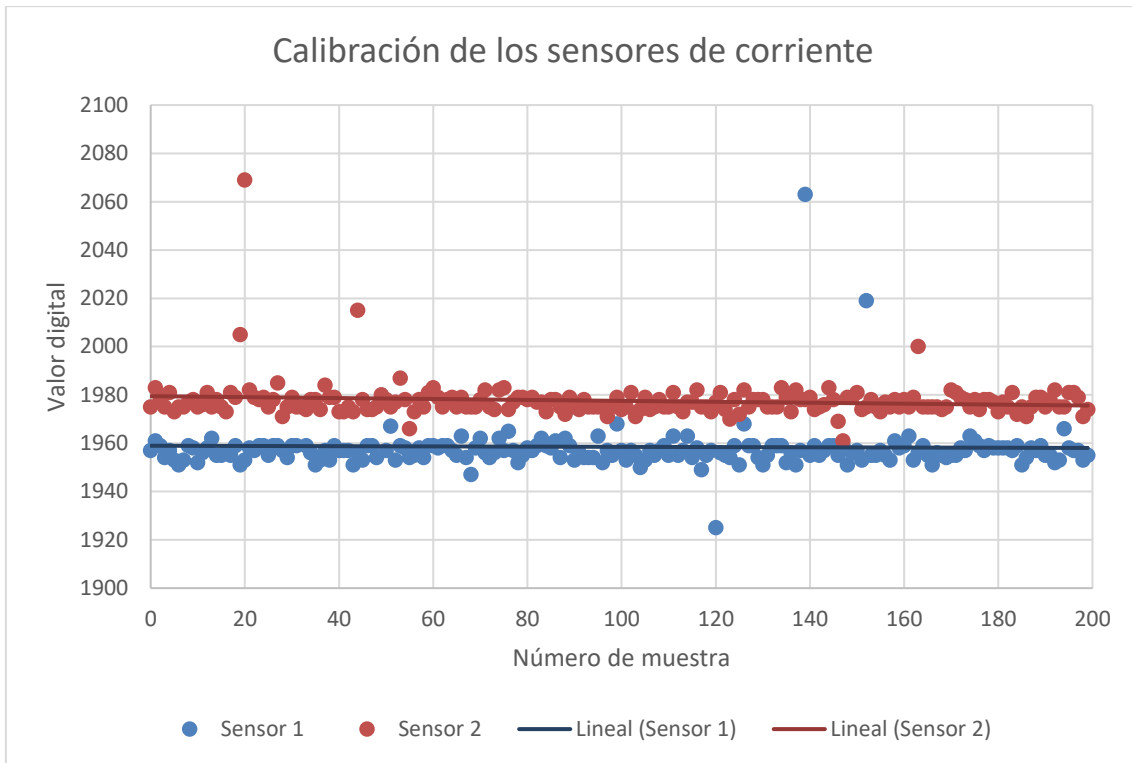
**Figura 90 Respuesta en frecuencia del filtro seleccionado (simulado con LT-Spice) (elaboración propia)**

Al implementar el filtro mencionado repetimos la prueba y observamos datos mucho más prometedores:



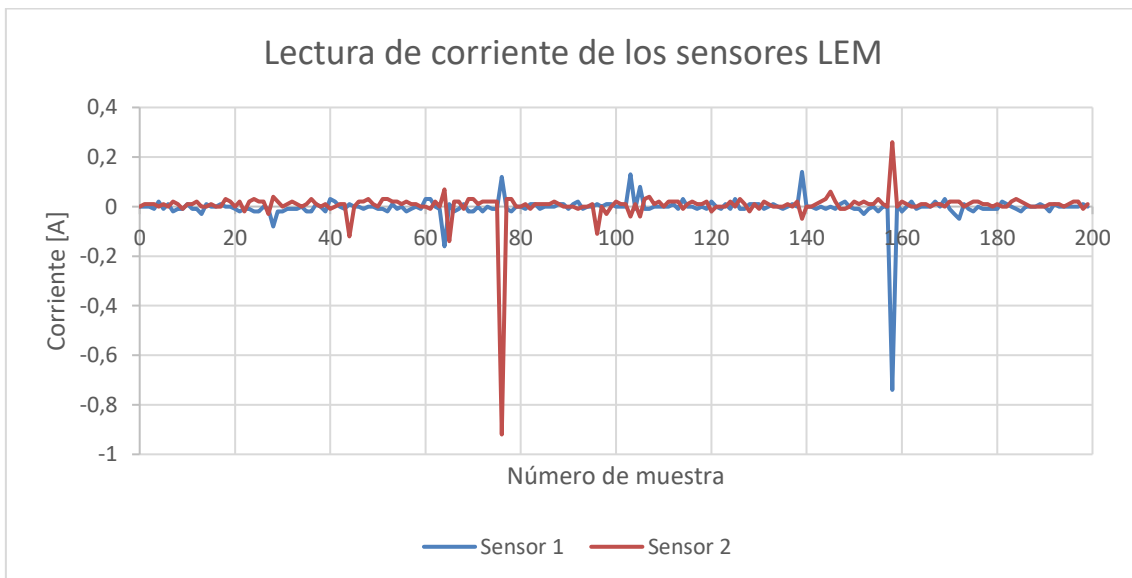
**Figura 91 Gráfica de corriente leída por el sensor LEM (filtrada) (elaboración propia)**

Podemos ver como ya no tenemos esa ondulación que observábamos antes, sin embargo, aún se pueden detectar dos errores, uno mucho más crítico que el anterior. El error principal es que ha una ligera variación para una corriente nula, vemos que aun sin comanda de tensión la corriente es ligeramente positiva, esto es debido a lo comentado anteriormente: la calibración se realizó con la alimentación proporcionada por el USB. Por ello, será necesario tomar los datos de lecturas de corriente nulas con el inversor proporcionando la alimentación y recalibrarlo.



**Figura 92 Calibración de corriente nula de los sensores de corriente (elaboración propia)**

Con los sensores recalibrados repetimos la prueba anterior y observamos lo siguiente:



**Figura 93 Lectura de corriente de los sensores LEM calibrados (elaboración propia)**

Ahora ya podemos ver que para ambos sensores la calibración sin corriente es correcta y, aunque observamos algo de ruido, este se podría eliminar sencillamente implementando un filtro digital en caso de que observemos que realmente causa problemas en el control de corriente.

### 3.11. CALIBRACIÓN DE LA ORIENTACIÓN DEL CAMPO DEL ROTOR

Lo último que debemos hacer antes de continuar con la implementación de los controladores es alinear el dato de posición que estamos leyendo mediante el encoder del servopack con la posición real del eje D que hemos definido como eje de corriente equivalente del imán permanente del rotor. Con esto se hace referencia a que el ángulo de posición que nos otorga el servopack no tiene por qué estar alineado con el eje D del motor, es decir con el vector de corriente equivalente del imán permanente.

Para alinearlos deberemos introducir un offset en la posición que leamos del encoder del servopack. Este offset lo podemos obtener de varias maneras. Una de las formas en las que lo podemos obtener es por comparación con la tensión de fase R del motor al girar con el ángulo medido por el servopack actualmente. Lo que haremos será generar una salida de tensión digital en forma de onda cuadrada mediante una de las salidas digitales del microcontrolador de forma que tenga un periodo de una revolución del motor, es decir que, si nuestro microcontrolador recibe 1000 pulsos por cada revolución del motor, nosotros intercambiaremos la salida entre nivel alto y nivel bajo cada 500 pulsos. Esta salida cuadrada y las tensiones de las diferentes fases las leeremos con el osciloscopio.

Lo que estamos esperando es detectar el cambio de positivo a negativo de la fase R. Ese punto debería coincidir si está alineado con el eje D. En la figura se representa esta situación en la que, al estar la corriente equivalente del imán permanente alineada con el eje d según lo hemos definido, al girar en sentido positivo, esa corriente del rotor produce una distribución de fems en los conductores del estator que es máxima positiva en la dirección del eje q negativo. Como en vacío en la fase R la tensión que se mide es justo la fem inducida cambiada de signo, en el instante en que el eje d del rotor está alineado con el eje de la fase R del estator, lo que para nosotros sería justo  $\lambda = 0$ , la tensión medida en la fase R debe ser 0 y estar cambiando de positiva a negativa, es decir, el paso por cero negativo de la tensión medida en la fase R:

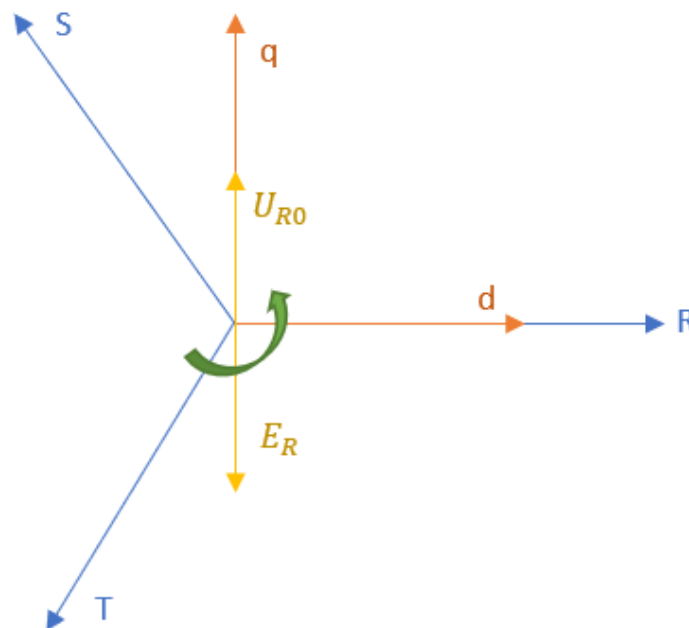


Figura 94 Diagrama vectorial de los ejes d y q junto con su sentido de giro (elaboración propia)

Para obtener el desfase de forma experimental se tomaron los datos a diferentes velocidades, en este caso desde 200 RPM a 1000 RPM con incrementos de 200 RPM cada vez. Recogidos los datos, mediante un análisis de Matlab se obtuvo la diferencia en grados sexagesimales que habría que desplazar la onda cuadrada generada por el microcontrolador hasta que esta coincidiera exactamente con el cambio de signo de la tensión de la fase R.

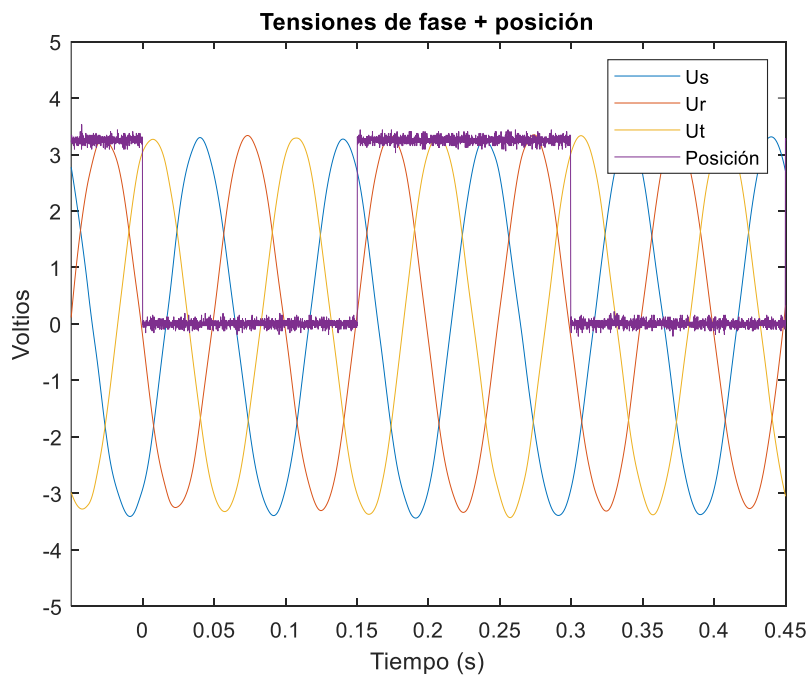
Además, al tratarse de un motor de 3 pares de polos, lo que observamos es que, para cada vuelta, lo que son 360 grados mecánicos, tenemos 3 vueltas eléctricas. Esto implica que, si nuestra medida de posición es correcta, dentro de la forma de onda cuadrada que estamos generando debemos encontrar 3 periodos de la onda de tensión de las fases.

Los datos de diferencia de posición que se obtuvieron fueron los siguientes:

Velocidad de giro (RPM)	Diferencia de ángulo (°)
200	236.60224
400	236.29813
600	235.91699
800	235.34087
1000	234.79247

**Tabla 2** Tabla de ensayos de desfase de ángulos

Se puede observar que la diferencia es consistente en todo el rango de velocidades. Una vez se ha obtenido la diferencia en posición volvemos a realizar la prueba y al visualizar los datos nos encontramos con lo siguiente:



**Figura 95** Comparación de las tensiones de fase con la señal de posición (elaboración propia)

Se observa exactamente lo que estábamos buscando, la onda cuadrada generada por el microcontrolador coincide con el cambio de signo de la fase R, en este caso representado por la forma de onda amarilla, y también se puede observar cómo existen 3 periodos exactos de la onda naranja ( $U_r$ ) dentro de la forma de onda cuadrada. En concreto, el flanco de bajada de la señal de posición, que se produce cuando la posición pasa de 999 a 0 cuando el rotor gira en sentido positivo (justo al inicio de una vuelta o  $\lambda = 0$ ) coincide exactamente con el paso por cero en sentido negativo, como habíamos establecido en relación a la figura anterior. Esto nos reafirma que la realimentación de posición funciona y que no estamos saltando pulsos del servopack.

Con esto comprobado, podemos proceder a la implementación de los controladores de corriente. Esto nos permitirá realizar las comprobaciones necesarias para saber si todo funciona.

Una de estas comprobaciones que nos dio problemas y se comentará más adelante fue justamente el mencionado alineamiento. El problema aparecía por una incoherencia entre lo que debemos esperar (si indicamos al controlador que queremos generar corrientes en el eje D y únicamente en el eje D el motor no debería de girar, ya que el eje D está alineado con la corriente equivalente del imán permanente y por lo tanto no estaríamos generando par). Sin embargo, en las pruebas iniciales de alimentación del eje d en lazo abierto que realizamos (sin control de corriente, simplemente aportando una tensión de eje d fija) vimos que el motor sí se movía al comandar corriente en el eje D. A raíz de esto se realizaron múltiples pruebas sobre el alineamiento, incluyendo la generación de la trama de pulsos para el inversor por comparación con la onda cuadrada anteriormente mencionada, para comprobar si todo funcionaba bien.

Finalmente resultó ser un problema relacionado con una de las funciones propias de la librería `math.h` que se utiliza en el código del microcontrolador y el orden de los argumentos que la función espera y los que se habían utilizado en el código de generación de tensión en los ejes (d,q). Como resultado de ese error, cuando se intentaba aplicar una tensión de eje d constante, en realidad lo que se aplicaba era una tensión de eje q constante, lo que hacía que circulara corriente de eje q y que el servomotor se pusiera en movimiento. Este es un ejemplo, dentro los muchos que se han descrito en el marco de este trabajo, que muestra la gran diferencia entre un diseño sobre el papel y un diseño que se va implementando paso a paso y que, en cada etapa, hay que reorientar para afrontar y resolver los problemas que surgen durante la implementación real.

A continuación, pasaremos a realizar el controlador de corriente y se comentarán las pruebas que se llevaron a cabo para obtener los parámetros del motor experimentalmente.

### 3.12. CONTROLADOR DE CORRIENTE

Comenzaremos con el código que se utiliza para el control de corriente. Este código corresponde a un controlador del tipo proporcional integral con implementación de límites y antiwind-up.

Para el control utilizaremos un paso de control fijo por programa de  $200\mu s$ , que en nuestro programa haremos referencia como  $dt$ .

Lo primero que necesitamos es definir cuál va a ser nuestra consigna, por supuesto esta vendrá dada por el usuario o, más adelante, desde el regulador de velocidad. Con la consigna definida pasamos a calcular el error cómo:

$$I_{error} = I_{consigna} - I_{leida}$$

Además de este error, debemos calcular cual será el error acumulado para la parte integral del controlador. Para esto, lo que haremos será sumar al error acumulado anteriormente el propio error por el paso de control fijo que hemos definido previamente, es decir, una integración numérica del tipo más sencillo (por rectángulos) típica de la implementación de controladores discretos en el tiempo:

$$I_{error_{acumulado}} = I_{error_{acumulado,previo}} \cdot dt$$

De esta forma el error acumulado irá aumentando con cada paso de control si existe error entre la lectura y la referencia dada.

Finalmente, la acción de control la calcularemos cómo:

$$V_{control} = Kp \cdot \left( I_{error} + \frac{I_{error_{acumulado}}}{t_i} \right)$$

Cabe remarcar que la acción de control será de tensión, y que la realizaremos sobre los ejes D y Q, por lo que en el código generaremos 2 controladores iguales para cada uno de los dos ejes.

Por último, debemos implementar también la limitación en la actuación, comúnmente conocido como sistema anti-windup. Para ello lo que haremos será limitar la salida a la vez que revertimos el error acumulado por la parte integral, de forma que no continúe almacenando error de forma indefinida si el controlador está saturado. Esto se realizará tanto si la saturación es positiva cómo si la saturación es negativa.

El código en nuestro caso implementado es el siguiente:



```

float ControlCorrienteD(float setpoint, float input)
{
    // Paso de control = 200uS
    float error = setpoint - input;
    erroracumuladod += error*dt;
    float output = Kp*(error + erroracumuladod/ti);
    if(output>=Vmax)
    {
        output = Vmax;
        erroracumuladod -= error*dt;
    }
    if(output<=-Vmax)
    {
        output = -Vmax;
        erroracumuladod -= error*dt;
    }
    return output;
}

```

Con el control de corriente generado, necesitamos saber cuáles son los parámetros que debemos introducir tanto de ganancia cómo de tiempo integral. Para ello haremos pruebas físicas sobre el motor que nos indicarán cual es tanto la resistencia interna de las fases como la inductancia del devanado. La necesidad de conocer estos parámetros radica en que, de acuerdo a la teoría de control de máquinas (cita mis apuntes de ADCAE: Apuntes de Análisis Dinámico y Control de Accionamientos Eléctricos, Javier Martínez Román, PoliformaT, UPV, 2022) el ajuste óptimo del regulador de corriente se debe conseguir ajustando un tiempo integral igual al tiempo característico del devanado en el que se desea controlar la corriente, y, para determinarlo, conviene determinar la resistencia e inductancia del devanado:

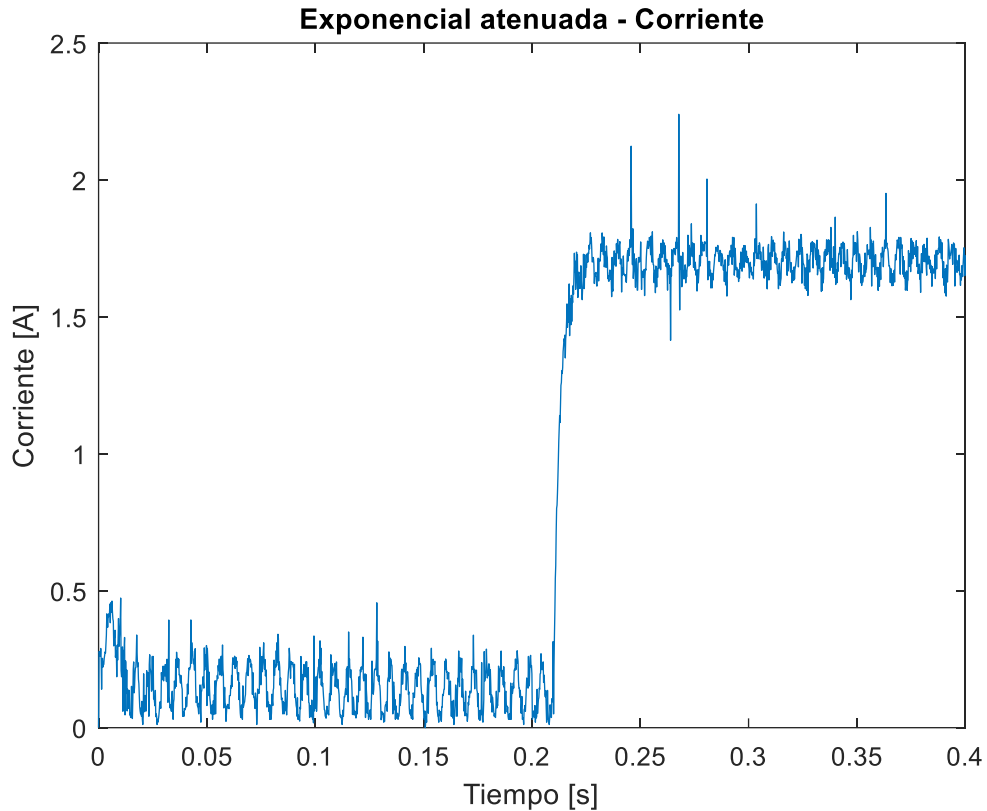
$$\tau_{dev} = \frac{L_{dev}}{R_{dev}}$$

Los datos de la resistencia del devanado son fáciles de obtener, lo que haremos será con ayuda de la fuente introducir una corriente conocida a través de 2 devanados y observaremos que caída de tensión se genera en las mismas. En nuestro caso hicimos pruebas a 1 amperio y obtuvimos una caída de tensión de 3V. Esto nos indica que la resistencia del devanado es de  $1.5\Omega$  ya que cómo se ha mencionado, la caída de tensión es de 2 devanados. La medida también se ha realizado con un multímetro y aunque menos precisa, también obtuvimos  $3\Omega$  de medida de resistencia en el devanado (a temperatura ambiente).

A continuación, realizaremos las pruebas necesarias para obtener la inductancia del devanado.

Para ello realizaremos una primera prueba con el controlador diseñado. Buscaremos obtener un salto de corriente en el eje D lo suficientemente alto cómo para, al tratarlo, ser capaces de ver cuál es la constante de tiempo propia del sistema. Esto es muy útil, ya que es un sistema sencillo del tipo LR como se comentó en la parte teórica, por lo que nos será fácil obtener a través de la constante de tiempo del sistema y la resistencia del devanado la inductancia del mismo.

Realizada la prueba lo que observamos es la repuesta que se recoge en la figura:



**Figura 96 Respuesta de la corriente ante un cambio de referencia (elaboración propia)**

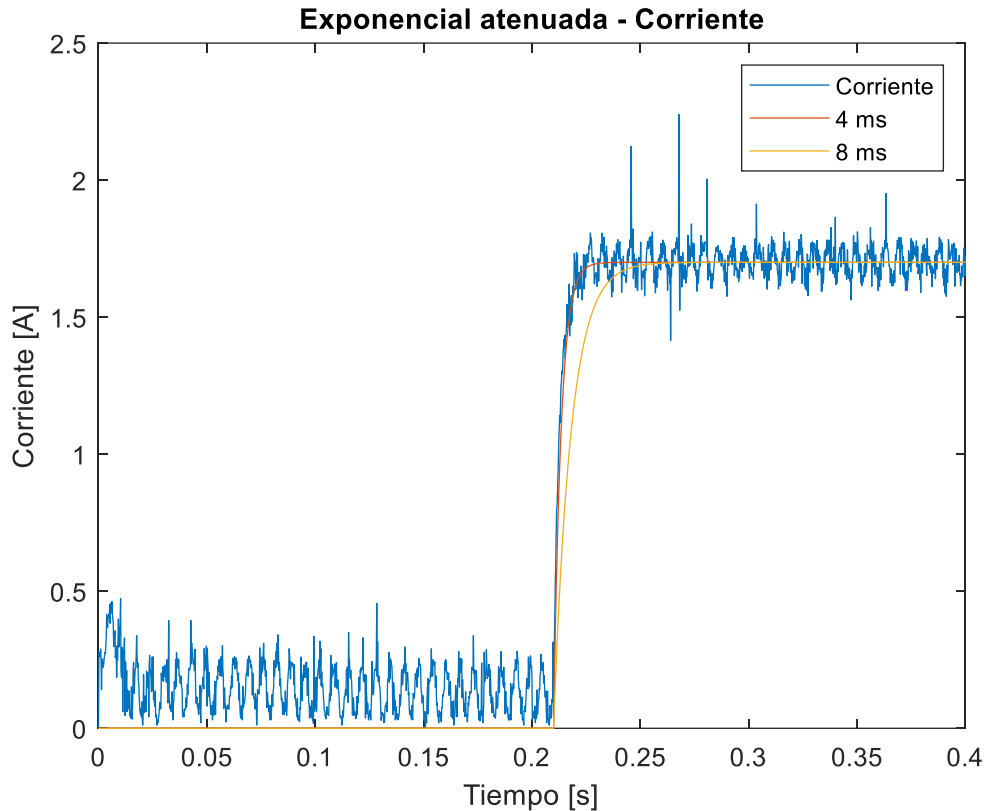
La gráfica representa el módulo del fasor de corriente generado.

Lo que haremos ahora será intentar superponer una exponencial atenuada de forma que siga lo mejor posible la curva de subida de corriente. Recordemos que el motor no está girando y que por lo tanto el circuito no es más que una inductancia en serie con una resistencia.

Este circuito tiene una expresión para la corriente:

$$I(t) = \frac{V}{R} \left( 1 - e^{-\frac{R}{L}t} \right)$$

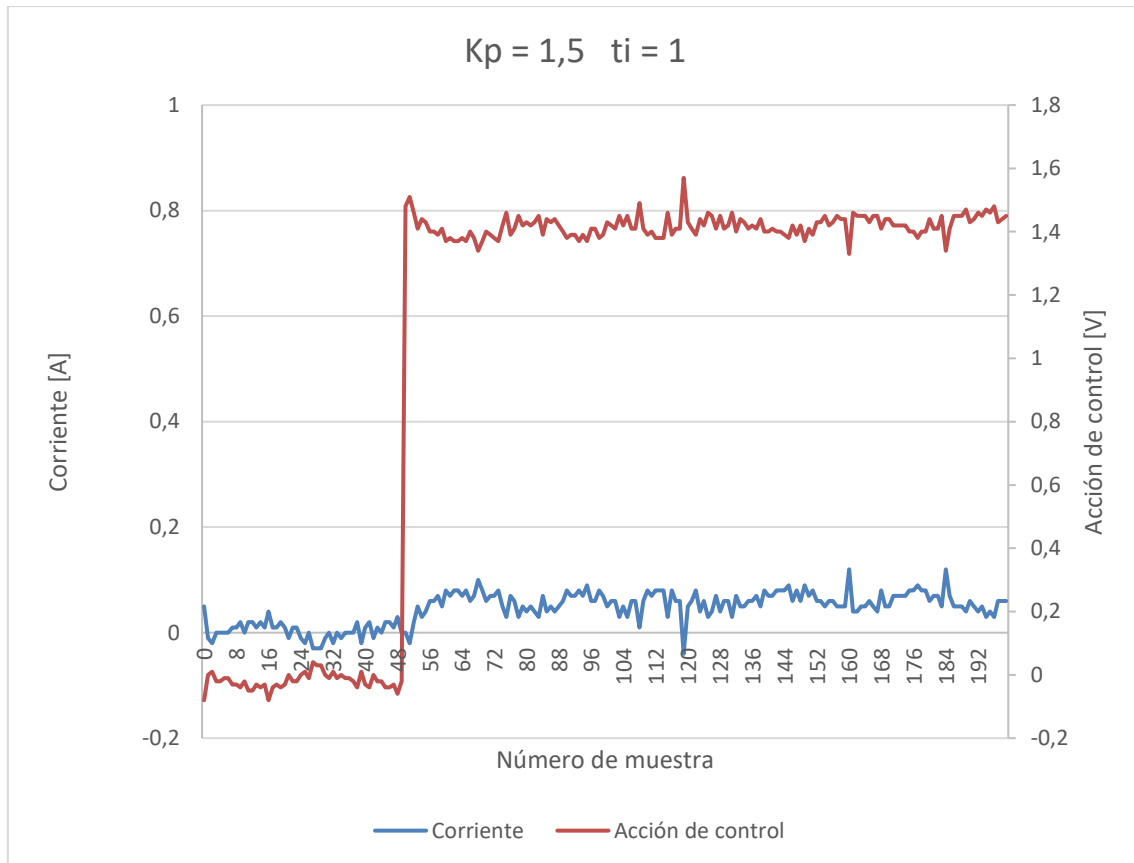
Sabemos que la resistencia del devanado son 1.5 ohmios, por lo que debemos de ir modificando la inductancia hasta encontrar una combinación que se acople de la mejor manera:



**Figura 97 Ajuste de exponencial atenuada a la respuesta real del sistema (elaboración propia)**

Esta curva exponencial se obtiene con un cociente  $\frac{R}{L}$  de 0.004. Lo que nos indica para una R de 1.5 ohmios que la inductancia es de 6mH. Esto, además, coincide con medidas tomadas años anteriores por en el departamento de Ingeniería Eléctrica (Román, 2008) en el mismo montaje con el mismo motor. Con estos datos obtenidos podemos comenzar a realizar las pruebas del control de corriente partiendo de una ganancia y tiempo integral que tienen un sentido físico. Dado que nuestro objetivo es mejorar todo lo posible la respuesta del controlador, iremos ajustando los valores paulatinamente hasta encontrar los que nos den una respuesta lo mejor posible alcanzando un compromiso adecuado según las reglas de diseño de controladores de corriente entre rapidez y estabilidad.

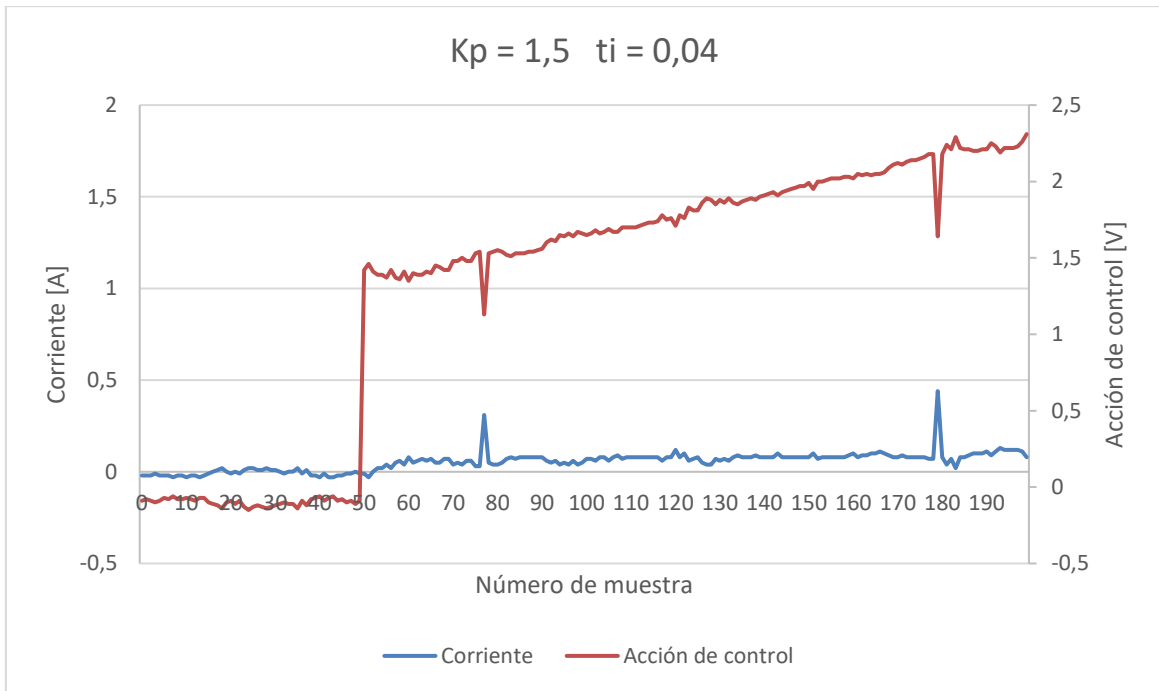
Realizaremos el primer ensayo con una Kp de 1.5 y un tiempo integral de 1 segundo, lo cual hace que prácticamente no exista acción integral.



**Figura 98 Respuesta del control de corriente ante un cambio en la referencia de tipo escalón (elaboración propia)**

Observamos que la respuesta de la corriente con respecto a la acción de control es muy baja. Además, le hemos indicado una corriente de referencia de 1A y, como podemos observar, no se llega a ésta en la duración de la prueba. También se observa que la acción de control se mantiene constante una vez aplicado el salto inicial, esto es por la falta de acción integral inherente del tiempo integral tan alto que le hemos indicado. El tiempo integral hace referencia a cada cuanto tiempo se duplicará la acción de control si se mantiene el error, y teniendo en cuenta que el eje X del gráfico son muestras de paso de control (que recordemos son 200 microsegundos) el tiempo de prueba que se está mostrando es de 0.04s, luego está claro que con 1 segundo de tiempo integral la acción de ésta es inapreciable.

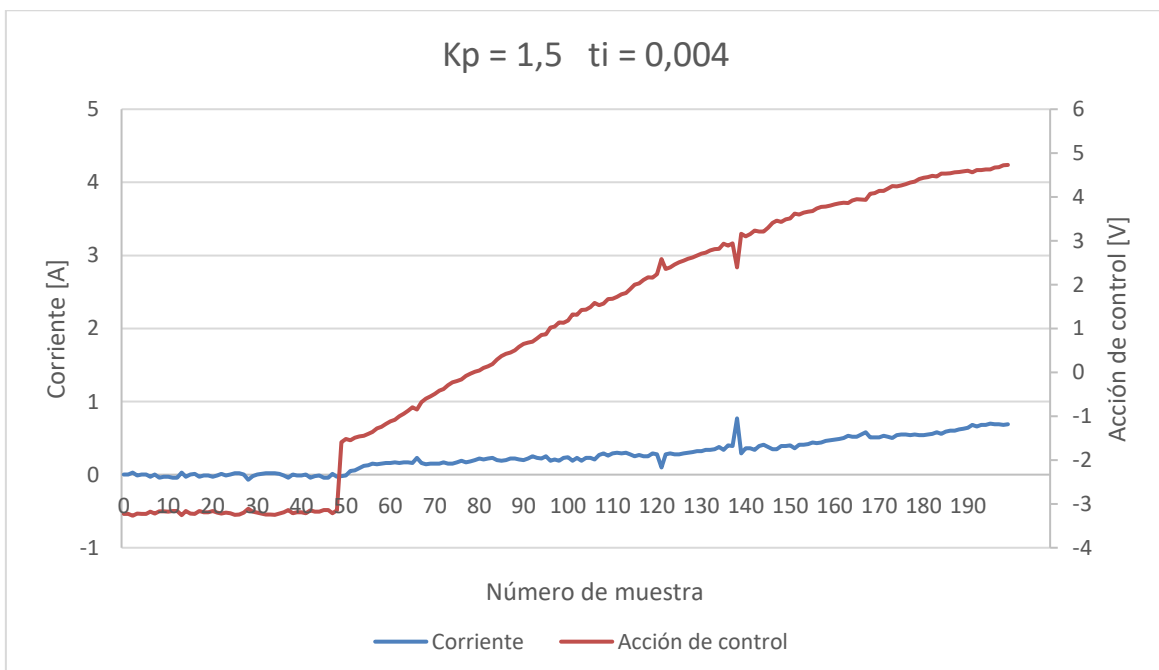
Observemos que es lo que ocurre cuando reducimos el tiempo integral a 0.04s:



**Figura 99 Respuesta del controlador ante un cambio de referencia en escalón con acción integral (elaboración propia)**

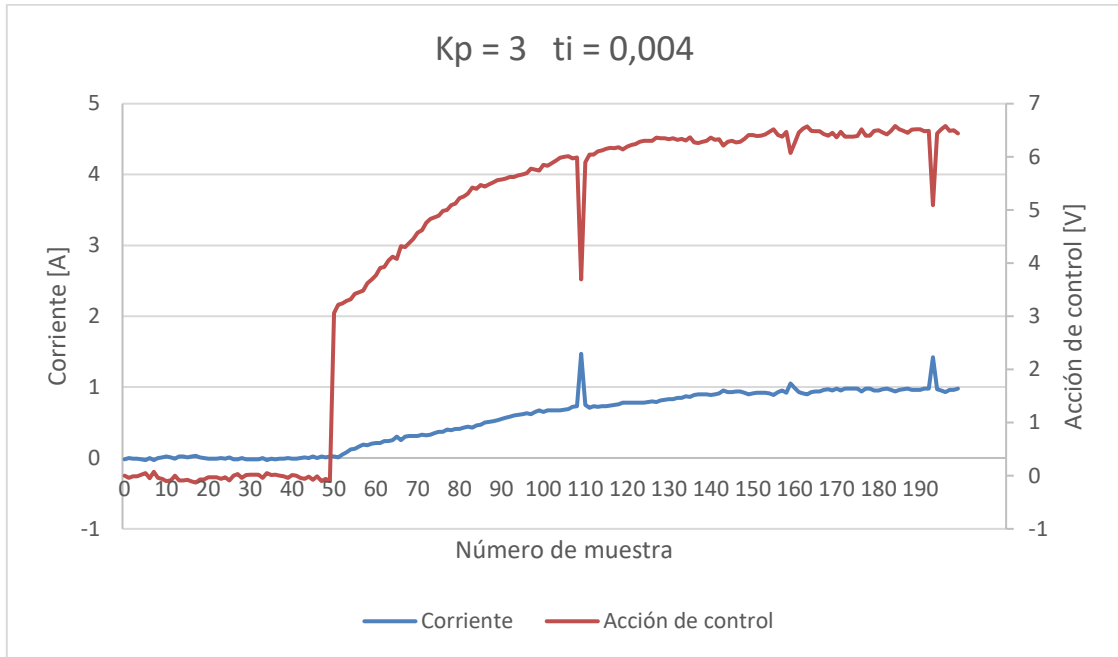
Podemos observar como en los 0.04s que ha durado el ensayo la acción de control se ha duplicado, lo cual nos remarca el funcionamiento correcto de la acción integral. Cabe denotar que los picos que observamos en los datos son debidos a ruido en la medida, originado por las conmutaciones del inversor.

Podemos continuar bajando el tiempo integral hasta el valor teórico que se calculó anteriormente, unos 4ms. Manteniendo la ganancia esto nos da una respuesta de esta manera:



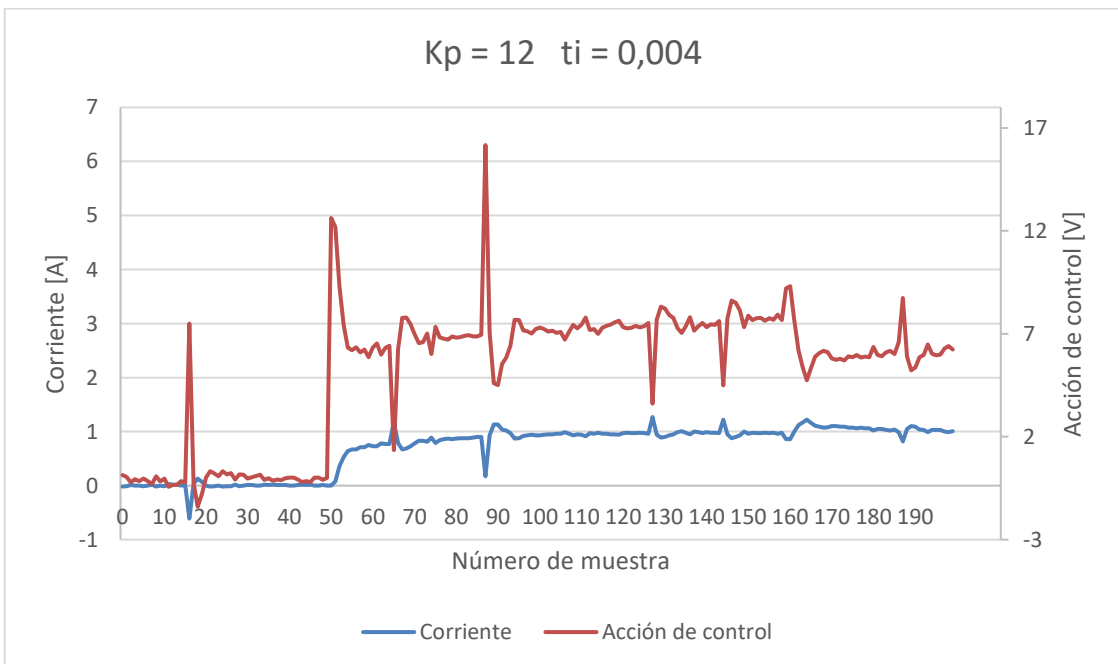
**Figura 100 Respuesta del controlador ante un cambio en escalón con acción integral actualizada (elaboración propia)**

Observamos que la acción integral ahora ya es muy notable y vemos como la corriente ya se va acercando al valor de referencia que se había indicado. Sin embargo, es notable que hace falta más ganancia para la parte proporcional, por lo que iremos aumentándola progresivamente.



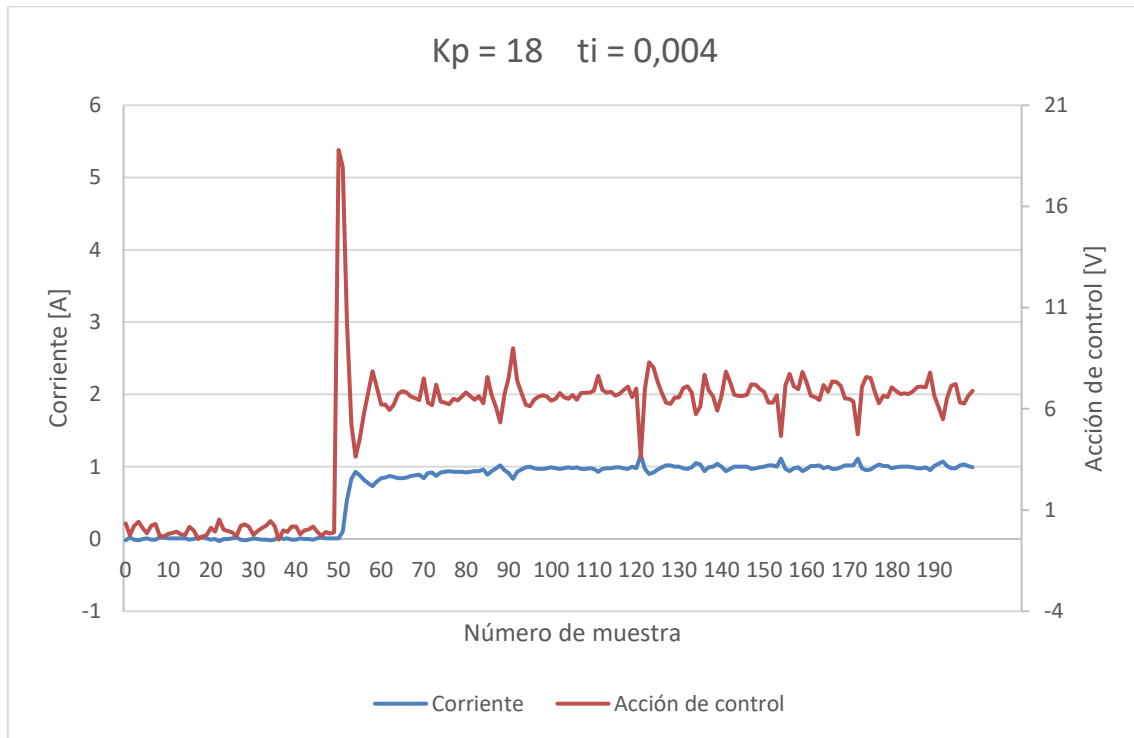
**Figura 101 Respuesta del controlador ante un cambio en escalón con aumento de ganancia (elaboración propia)**

Aumentando la ganancia podemos ver como en los 40ms que dura la prueba ya alcanzamos el valor de referencia de la corriente, sin embargo, lo que buscamos con esta prueba no es eso, buscamos el punto a partir del cual la respuesta de la corriente comienza a tener un valor subamortiguado.



**Figura 102 Respuesta del controlador ante cambio en escalón con gran aumento en la ganancia (elaboración propia)**

Vemos que con una ganancia significativamente mayor observamos una respuesta en corriente muy buena, y, de hecho, si seguimos aumentando la ganancia observamos justo lo que esperábamos, encontramos esa respuesta subamortiguada que, realmente, queremos evitar (porque no incrementa la rapidez de respuesta, sino que hace que el tiempo de establecimiento sea incluso mayor debido a la sobreoscilación, además de acercarnos al comportamiento oscilante):



**Figura 103 Respuesta final del controlador de corriente ante un cambio en escalón (elaboración propia)**

Con esto podemos dar por zanjado el control por corriente, y lo único que nos quedará será, con este control acabado, implementar el control de velocidad.

Para realizar el control de velocidad lo primero que necesitaremos es ser capaces de medir la velocidad del motor. Para esto haremos uso de la medida de posición para obtener la velocidad a partir de ésta.

### 3.13. OBTENCIÓN DE LA LECTURA DE VELOCIDAD DEL EJE

Para la obtención de la velocidad del eje, como se ha comentado previamente, realizaremos una estimación a partir de la posición. Una primera idea podría ser simplemente utilizar el incremento entre posiciones y el paso de control que conocemos entre ellas para extrapolar la velocidad del eje. Esto, aunque aparentemente buena idea, lo que nos provoca es mucha dispersión en la medida, ya que ante los propios cambios rápidos del giro del eje la medida tiene una alta variabilidad, que la vuelve inservible para implementar un control de velocidad efectivo, lo veremos más adelante.

La forma en la que lo haremos, será tratando de estimar la posición actual partiendo de un cálculo de velocidad. Esto lo que quiere decir es que, a través de la posición real que hemos leído en una primera iteración calcularemos un error:

$$error = posición_{actual} - posición_{estimada}$$

La velocidad se calculará ponderando con este error y una ganancia que ajustaremos más adelante mediante pruebas y ensayos a velocidades conocidas:

$$velocidad = error \cdot K_v$$

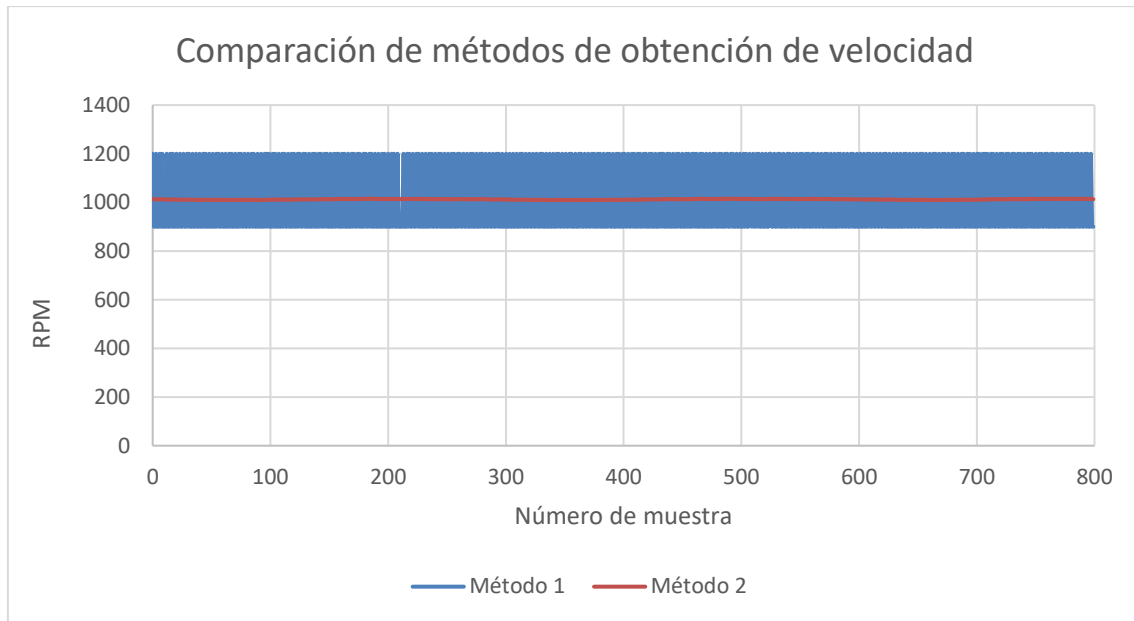
Con la velocidad estimada, volvemos a calcular la posición estimada con ayuda de la velocidad que acabamos de calcular y el paso de control de nuestro sistema, que llamamos dt:

$$posición_{estimada} = posición_{estimadaanterior} + velocidad \cdot dt$$

La posición estimada anterior nos hace falta porque recordemos que la posición es un valor que en nuestro caso hace referencia a tantos pulsos por revolución como se configuren en el servopack.

Mediante esta sucesión de cálculos para estimar la posición lo que vamos haciendo es modificar nuestra estimación de la velocidad para darnos un valor mucho más estable que el que nos podría dar el cálculo por derivada numérica de la posición respecto del tiempo. Además, esta estabilidad se puede ponderar gracias a la ganancia  $K_v$  que hemos mencionado anteriormente. Cuanta mayor es la ganancia de velocidad mejor es la respuesta dinámica y menos retrasos secundarios tendremos, ya que en menos iteraciones llegaremos a un valor sensato de velocidad, pero mayor dispersión tendremos en la medida. Veamos una comparación con la estimación de la velocidad por derivada numérica de la posición:





**Figura 104 Comparación de métodos de obtención de velocidad (elaboración propia)**

El método 1 es el primeramente mencionado, que, como podemos observar, para una velocidad real de 1000 RPM nos otorga mucho ruido en la medida, sin embargo, el método 2, el anteriormente expuesto observamos que nos da un resultado fantástico sin apenas ruido en la medida.

Una vez tenemos la realimentación de velocidad controlada podemos pasar a realizar el control de velocidad final del motor.

### 3.14. CONTROL DE VELOCIDAD

Llegamos finalmente al verdadero objetivo de un proyecto como este, realizar el control de velocidad del motor. Lo que buscamos es poder definir una velocidad objetivo o de referencia y que, a través del control que hemos ido realizando lleguemos a esta velocidad de referencia con una respuesta dinámica lo más adecuada posible a las posibilidades que tenemos.

Para esto, lo primero que debemos hacer es implementar el controlador de velocidad de tipo proporcional integral en nuestro microcontrolador. Este controlador será del mismo estilo que el generado para el control de corriente. Su acción de control en este caso será la corriente en eje q (la que genera par).

Igual que en cualquier controlador de este estilo, necesitaremos proporcionar una ganancia para la parte proporcional y un tiempo integral para la parte, obviamente, integral.

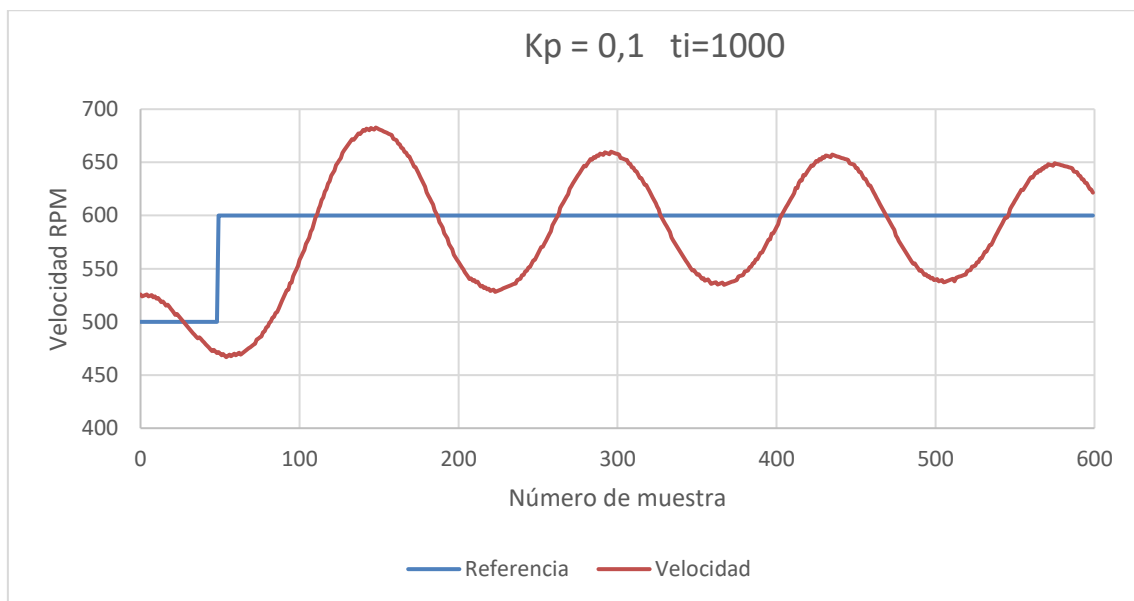
Además, igual que en el control de corriente, añadiremos un sistema de límites con anti-windup que evite que el controlador demande excesiva corriente, en nuestro caso no más de 3A continuamente y que contrarreste la acumulación de error por parte de la parte integral del controlador en el funcionamiento saturado con una salida limitada.

Como vemos, el controlador es del mismo estilo que los controladores de corriente que creamos anteriormente. En este caso la referencia de velocidad será lo que nosotros le introduciremos como parámetro.

Lo que necesitamos realizar a continuación, es lo mismo que realizamos en la parte teórica del control de velocidad, ajustar el controlador. El procedimiento clásico de ajuste se basa en una secuencia en dos etapas. Primero se aumenta la ganancia del controlador hasta que observemos un comportamiento amortiguado crítico o con una ligera sobreoscilación en la respuesta, mientras se mantiene la acción integral prácticamente nula. En segundo lugar, una vez encontremos un punto en el que estemos contentos con la respuesta dinámica (amortiguamiento crítico o ligeramente subamortiguado) sin acción integral, continuaremos aumentando la intensidad de respuesta de la parte integral (reduciendo el tiempo integral). Aumentaremos esta acción integral hasta que veamos que la respuesta dinámica del controlador comienza a empeorar, será en este punto en el que demos por concluido el diseño del controlador. De esta forma conseguimos el valor más bajo aceptable de tiempo integral que supone una respuesta con cancelación rápida del error ante una entrada en escalón y, además, el menor error de seguimiento posible en respuesta a rampa de consigna de velocidad.

En este caso sin embargo no será tan fácil como en la teoría. Esto es debido a que todos los pasos de control que hemos ido implementando aplican retrasos secundarios, que causan que el sistema deje de ser de primer orden y pase a ser de orden superior, lo cual empeora la respuesta y complica la tarea de diseñar el controlador.

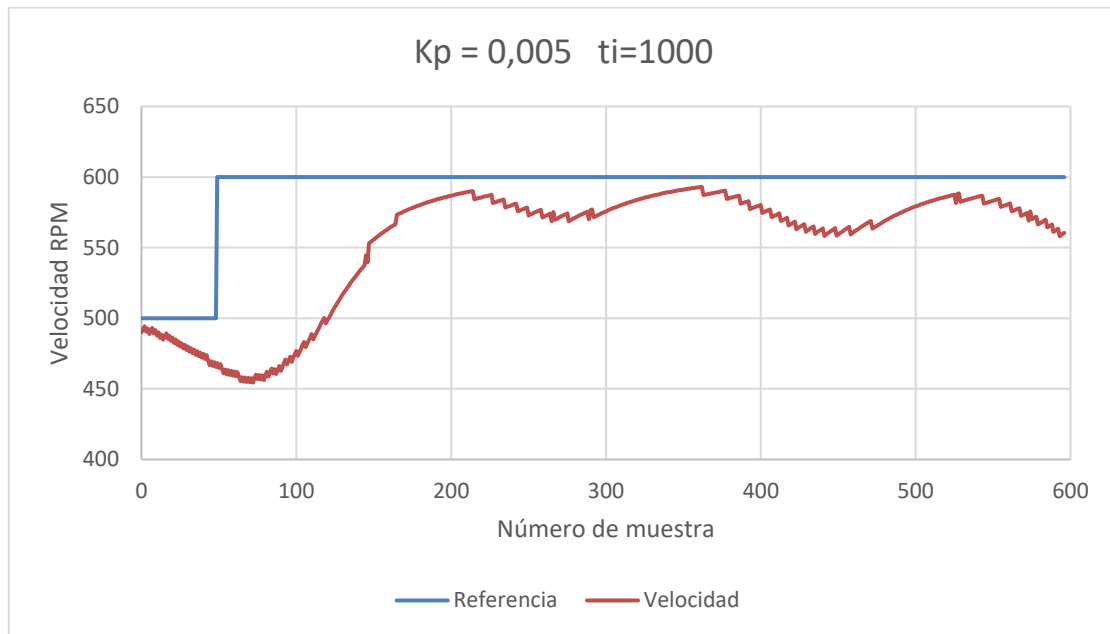
En los ensayos comenzamos por una ganancia que, en su momento creíamos sería lo suficientemente baja como para no provocar sobreoscilaciones, de un valor de 0.1. Sin embargo, como vemos en el gráfico siguiente, esta ganancia es muy superior a la que buscamos y podemos observar las grandes sobreoscilaciones que esto causa:



**Figura 105 Respuesta del control de velocidad con clara oscilación ante cambios en la referencia de tipo escalón (elaboración propia)**

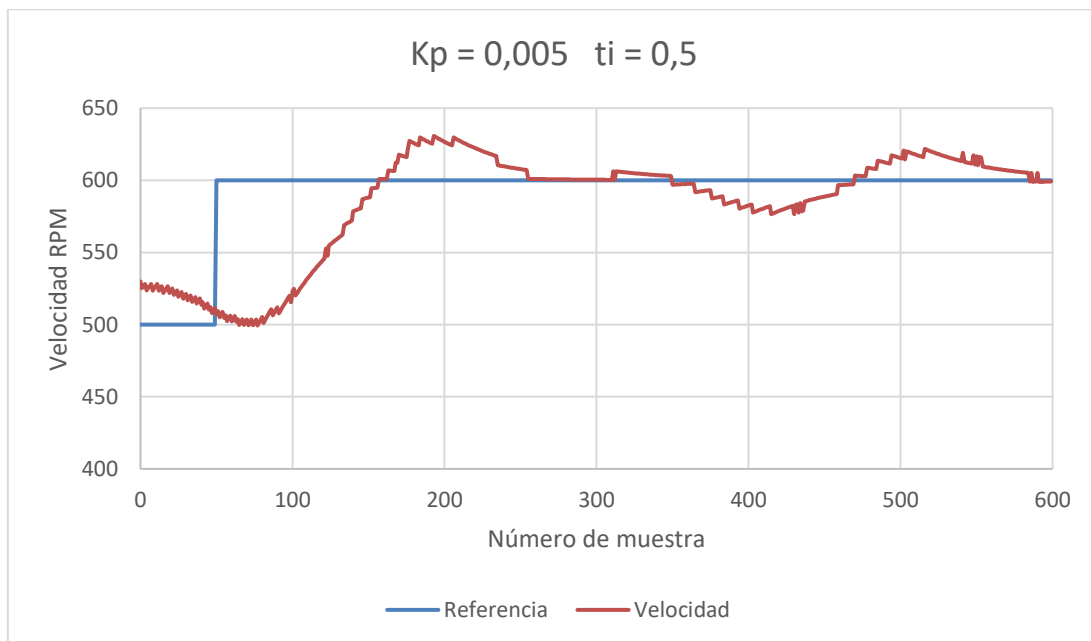
Cabe remarcar que el salto de velocidad que se le demanda al controlador es de un valor bajo, algo que pueda realizar rápidamente y cuya respuesta dinámica afecte lo mínimo posible al control sin llegar a trabajar saturado, es decir, con su salida limitada (es importante hacer el ajuste de la ganancia en estas condiciones para evitar introducir no linealidades en la respuesta, asociadas a la saturación de la salida del regulador, que sólo dificultarían aún más el ajuste del

regulador). Observando que la ganancia es claramente demasiado alta, lo que haremos será reducirla un orden de magnitud y repetir el ensayo:



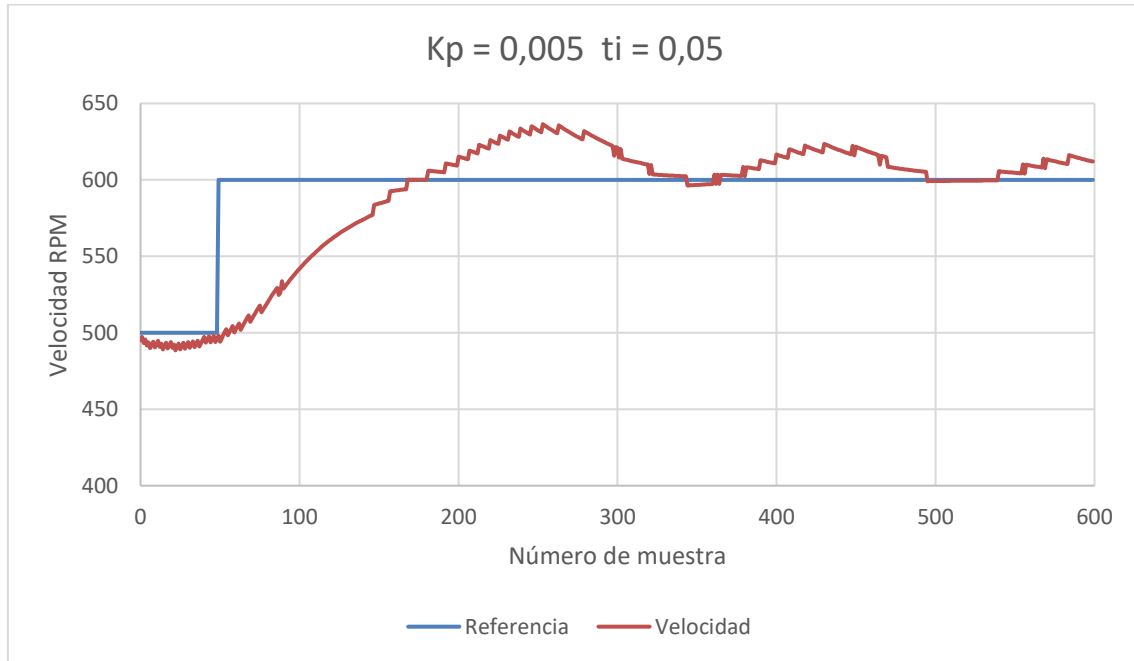
**Figura 106 Respuesta mejorada del controlador sin acción integral (elaboración propia)**

En este ensayo se puede apreciar en mayor medida las pequeñas oscilaciones causadas por un aumento en la ganancia en el cálculo de la velocidad (recordemos que cuanto mayor sea la ganancia, menores son los retrasos secundarios, pero más oscilación captaremos en la medida). Adicionalmente, podemos observar una gran mejora en la sobreoscilación de la velocidad angular, por lo que en adelante podemos comenzar a implementar la parte integral del controlador de velocidad. Comenzaremos con un tiempo integral relativamente alto, de unos 500ms, lo que prevemos observar es una oscilación similar con un ajuste a los valores de referencia mucho más pronunciado (aportado por la propia acción integral).



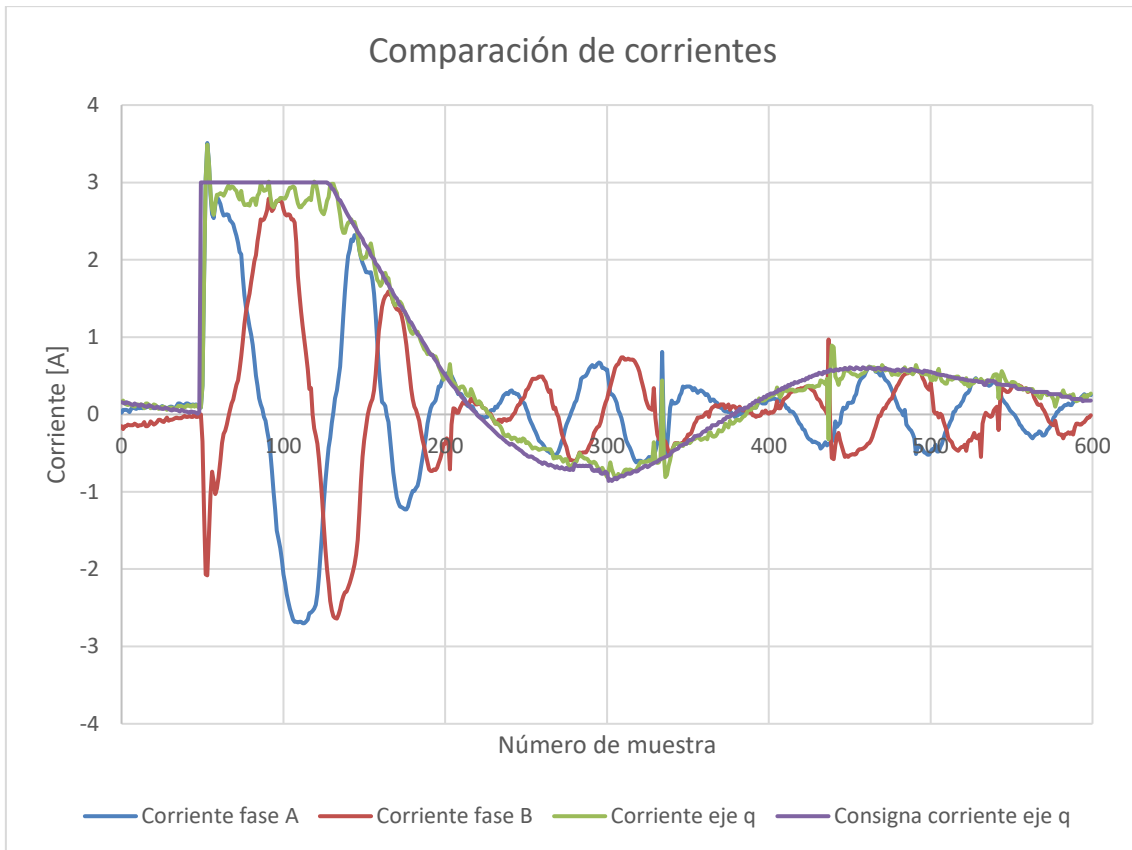
**Figura 107 Respuesta mejorada del controlador con acción integral (elaboración propia)**

Cómo observamos, no alcanzamos a ver ningún cambio en la sobreoscilación de la velocidad, por lo que continuaremos rebajando el tiempo integral hasta un valor que consideremos favorable, un buen valor al que apuntar sería reducir el tiempo integral en un factor de 10, aumentando su efecto así en la misma magnitud. En caso de que este ensayo nos aporte un resultado similar al que observamos en este anterior (es decir, que la respuesta no empeore) podemos dar por completado el ensayo del controlador de velocidad y con él, el diseño del control de velocidad del motor:



**Figura 108 Respuesta del controlador con aumento de la acción integral (elaboración propia)**

Apenas se observa un cambio significativo en la respuesta del controlador, por lo que cómo se ha comentado, podemos dar por concluido el diseño del controlador de la máquina de imanes permanentes síncrona. Adicionalmente, podemos realizar un último ensayo, tomando los datos de consigna de corriente de eje q, corriente de eje q y corrientes en las fases A y B del motor:



**Figura 109 Comparación de la respuesta del controlador de corriente ante un cambio en la consigna de velocidad (elaboración propia)**

En la figura se observa perfectamente el ajuste adecuado del lazo de corriente de eje q, que se traduce en un valor actual que sigue con gran rapidez y precisión la corriente de eje q demandada, el correcto control de corriente por orientación por campo del rotor, que produce corrientes de fase de variación senoidal con amplitud modulada por la consigna de corriente de eje q, y también, de forma indirecta, el correcto funcionamiento del control de velocidad que asegura una aceleración del valor máximo permitido por el motor (corriente de eje q máxima y plana durante la aceleración) seguida de un rápido ajuste de la velocidad con una ligera sobreoscilación marcada por la corriente de eje q de sentido negativo (necesaria para frenar un poco el motor tras la sobreoscilación de la velocidad).

## **4. DISEÑO DE LA TARJETA DE CONTROL PARA FABRICACIÓN EN CIRCUITO IMPRESO**

### **4.1. INTRODUCCIÓN**

Como complemento al proyecto se tomó la decisión de realizar el diseño de una tarjeta de control de especificaciones superiores a la diseñada como prototipo para su uso complementario con el equipo. Las especificaciones de la tarjeta son similares a las especificaciones fijadas para el propio proyecto, con la adición de un sistema de alimentación aislado para las señales de control y del inversor (para contar con un mayor factor de seguridad en la tarjeta) y de una conexión aislada mediante USB, que permita la conexión de la tarjeta al ordenador simultáneamente con el inversor. Esto no es posible en la tarjeta de prototipado que se montó en el laboratorio ya que el inversor trabaja a una tensión muy elevada, la cual crea una diferencia de potencial entre el terminal negativo o “GND” del inversor (y por lo tanto de la tarjeta de control) y la tierra de protección o masa a la cual están conectadas las partes expuestas del inversor.

### **4.2. TECNOLOGÍA DE FABRICACIÓN DE LA TARJETA DE CONTROL**

La tarjeta de control se ha diseñado para ser fabricada con tecnología SMD (tecnología de montaje superficial por sus siglas en inglés) en una placa de circuito impreso o PCB de 4 capas. Esta decisión viene condicionada por varios motivos: el primero y más importante es el tamaño de los componentes y la disponibilidad; al no trabajar con módulos de evaluación sino con los propios componentes sueltos muchos de ellos no tienen versiones THT (through hole technology o tecnología de orificio pasante por sus siglas en inglés), esto implica la necesidad de utilizar componentes SMD inevitablemente, lo cual condiciona el diseño completo de la placa. El segundo motivo para la justificación del uso de 4 capas en la tarjeta de control es para mejorar la compatibilidad electromagnética de la tarjeta. Tal y como se ha expuesto en el primer capítulo del proyecto, el ruido electromagnético generado por el inversor y por el servopack causa graves problemas en las lecturas de posición y corriente, adicionalmente, al trabajar con un microcontrolador diseñado para aplicaciones IoT este está preparado para trabajar con señales electromagnéticas de baja intensidad y su exposición a señales similares de más intensidad puede afectar al desempeño de las comunicaciones como el Bluetooth, que en nuestro caso es vital para la comunicación entre el control y la tarjeta.

El uso de aisladores para la conexión USB también fue un requisito importante para poder facilitar la conexión de la tarjeta al ordenador mientras ésta está conectada al inversor. Por motivos de compatibilidad y sencillez de uso se optó por el montaje de dos conectores USB, uno con capacidad de aislamiento hasta 5000V y otro sin aislamiento, en caso de que uno de los dos fallara la comunicación con la tarjeta no tendría por qué desaparecer por completo.

Para el aislamiento del USB se optó por un ADuM4160 (Devicies), un aislador de conexión USB 2.0 con hasta 12 Mbps de velocidad máxima y con tensiones de aislamiento de hasta 5000 Vrms. Los conectores USB utilizados fueron de tipo USB-C para maximizar la compatibilidad con todo tipo de dispositivos más modernos y la alimentación de la tarjeta se puede hacer vía USB (5V) junto con un regulador a 3.3V para el microcontrolador o vía un conector externo a 12V, ya que

la tarjeta incorpora el diseño de un DC-DC tipo Buck que rectifica los 12V hasta la línea de tensión de 5V a la que trabaja el USB. Cabe remarcar que las distintas alimentaciones están aisladas mediante diodos las unas de las otras; de esta forma se previene el hecho de realimentar el conector USB a través del DC-DC de conexión externa.

Adicionalmente, se optó también por un diseño modular en cuanto al montaje de las puertas lógicas. Esto se hizo con motivo de maximizar la compatibilidad de la tarjeta con distintos modelos de inversores y de montaje que requirieran de puertas del tipo negadas. Para ello se optó por el montaje de un pequeño selector por conector puenteado y de un zócalo de 14 pines que permita el intercambio entre integrados con distintas puertas lógicas.

Por último, remarcar que también se realizó el diseño del circuito complementario para la programación del microcontrolador. Este circuito, compuesto por un CP2102, es un adaptador de señales USB a comunicación serial. En la placa también se añadieron distintos puntos de testeo para poder realizar las mismas pruebas realizadas en el prototipo de forma sencilla con conectores apropiados para las sondas de osciloscopio.

### **4.3. STACK UP DE LA TARJETA**

En cuanto al stack up de la tarjeta se optó por un diseño que combina señales en las capas superior e inferior de la tarjeta, junto a un plano de masa (con separación entre los dos planos aislados) en segunda posición, justo debajo de las señales de la capa superior, donde se montan los componentes, y un plano de potencia en la tercera capa separando las distintas líneas de tensión por grupos.

El realizar el stack up de esta forma nos ayuda a la prevención de la incorporación de ruido electromagnético en las señales. Adicionalmente, se ha tenido muy en cuenta en el diseño las impedancias de señales como la de USB, que adicionalmente, también es una señal diferencial, por lo que se ha de ajustar el diseño para minimizar el ruido al que este par está expuesto.

Los planos de los distintos esquemáticos de la tarjeta se encuentran en el capítulo pertinente a planos, junto con el diseño de la propia tarjeta, separado por capas.





## 5. CONCLUSIONES

A modo de resumen, se ha realizado en primera instancia un análisis teórico profundo del funcionamiento de un motor síncrono de imanes permanentes, con el objetivo de comprenderlo en profundidad y de simplificar el posterior diseño del controlador.

Con los conceptos claros del funcionamiento del motor se comenzó con el estudio de la realimentación de posición del eje del motor. Para ello se optó por hacer uso del encoder que el servopack al que estaba conectado el motor acoplado al servo que se manejará con la tarjeta. Esta decisión condicionó inicialmente el proyecto, debido a la gran cantidad de ruido que el servopack introducía en las señales de posición. Para tratar de subsanarlo se optó por un optoacoplador con salida transistor, que resultó ser demasiado lento para las velocidades a las que se planeó que el motor fuera a funcionar. Este error se subsanó utilizando optoacopladores con salida digital, realmente concebidos para su uso como aisladores en señales de potencia pero que debido a la escasez de componentes y a los largos tiempos de entrega de los mismos nos fueron claves por ya contar con ellos en el laboratorio. Se continuó con el diseño teórico del controlador, generando un equivalente teórico del motor en Simulink e introduciendo en el equivalente teórico las realimentaciones de corriente y posición pertinentes para su funcionamiento. Se realizó el diseño del control de corriente, ajustando los controladores de forma teórica y comprobando su validez en base a simulaciones continuas del mismo. Una vez diseñado el controlador de corriente se continuó con el diseño del controlador de velocidad, de igual manera se diseñó el control para obtener la mejor respuesta posible en base a las simulaciones realizadas y se comprobaron los métodos de ajuste de los controladores para agilizar su posterior implementación en el código del microcontrolador.

Una vez la parte teórica estuvo acabada, se pasó al montaje de la tarjeta prototipo junto con los sensores LEM de corriente. Se realizaron las pruebas de lectura de corriente y su calibración y se pasó a la implementación y las pruebas del controlador de corriente. Con éste funcional se implementó el controlador de velocidad de igual manera. Paralelo al diseño de los controladores también se implementó un sistema de comunicación Bluetooth a través de una tableta y una aplicación genérica de puerto serie para la comunicación aislada entre la tarjeta y el usuario, así como para la recogida de los datos de los ensayos realizados.

Por último, se realizó un diseño de un circuito impreso para la posible fabricación industrial de una tarjeta con características mejoradas a la tarjeta prototipo montada en el laboratorio. Se generaron los documentos pertinentes para la posibilidad de enviar a cualquier fabricante externo y posibilitar la fabricación en serie de la tarjeta.

El banco de ensayos desarrollado y comprobado en el marco de este trabajo cumple, por tanto, con los dos objetivos principales establecidos en la introducción: por una parte, se ha conseguido un diseño efectivo y de muy buenas prestaciones para el control de servomotores de imanes permanentes de técnica senoidal (incluyendo como elementos fundamentales el diseño de la realimentación de posición, del conjunto del sistema de control, y de una tarjeta de control de características industriales) y, por otra, en el diseño se ha incorporado como característica fundamental la orientación al aprendizaje de las técnicas de accionamientos eléctricos (implementando el control al 100% sobre software, lo que lo hace completamente

flexible, e incorporando la posibilidad, puesta de manifiesto en los cientos de ensayos realizados, de acceder a todas las entradas, salidas y parámetros de control de forma sencilla – y muy cómoda además, mediante una interfaz inalámbrica sobre dispositivo móvil).

## 6. REFERENCIAS

AKM. (s.f.). *Structure of optical encoder*.

Devicies, A. (s.f.). *ADum4160 Datasheet*.

Diagram, S. (s.f.). *Basic Structure of PMSM ResearchGate*.

LEM. (s.f.). *LTS 6-NP Current Transducer Datasheet*.

Lerma, A. P. (s.f.). *Documentación PMSM*.

LiteON. (s.f.). *Photocoupler Product Data Sheet*.

própia, E. (s.f.). *Elaboración própia*.

Román, J. M. (2008). *Ensayos de Servomotores Yaskawa SGDh por el departamento de ingeniería eléctrica, Universidad Politécnica de Valencia*.

Yaskawa. (2003).  *$\Sigma$ -II Series SGMH User Manual*. Yaskawa.

Yaskawa. (2003).  *$\Sigma$ -II Series SGMH User Manual*.





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIERÍA  
INDUSTRIAL VALENCIA

**DISEÑO DE UN SISTEMA DE CONTROL DE INVERSOR  
PARA UN SERVO SENOIDAL DE 200W**

---

**PRESUPUESTO**

---

Trabajo Final de Grado

Antonio Poveda Serrano

Julio, 2022

## **PRESUPUESTO**

A continuación, se expone el documento que pretende enumerar en términos económicos la elaboración del proyecto en cuestión.

El presupuesto ha sido subdividido en 3 capítulos de forma que los dos primeros tendrán como objetivo la descripción monetaria de la elaboración física del proyecto y el último la redacción de la presente memoria. Además, en cada capítulo se ha subdividido el coste en material y humano.

Teniendo en cuenta el objetivo del proyecto, la elaboración de un control de máquinas síncronas de imanes permanentes mediante un microcontrolador ESP32 (**o descripción similar**), el coste económico sobre el que repercute el proyecto incluye los componentes electrónicos necesarios para el montaje de la tarjeta de control, así como sus respectivos conectores y cableados para su uso en el laboratorio. Como referencias se ha consultado el catálogo de las empresas *Mouser Electronics*, *Farnell*, *Digi-key* y *RS Components*.

CAPÍTULO 01 TARJETA DE CONTROL				
RECURSOS	CONCEPTO	CANTIDAD	PRECIO UNITARIO	COSTE
Materiales	ESP32 DevKit V1	1 u.	8.42 €/u	8.42 €
	Optoacoplador Digital HCPL-3120	3 u.	4.74 €/u	14.22 €
	Puerta OR SN74HC32DR	1 u.	0.38 €/u	0.38 €
	Sensor LEM Corriente LTS6-NP	2 u.	15.59 €/u	31.18 €
	Cabezal de 40 pines rectos con paso 2.54mm	1 u.	0.525 €/u	0.525 €
	Cabezal de 6 pines rectos con paso 2.54mm	1 u.	0.32 €/u	0.32 €
	Funda termorretráctil	0.1 m	0.50 €/m	0.05 €
	Caja de conexiones	1 u.	3 €/u	3 €
	Tornillería	4 u.	0.39 €/u	1.56 €
	Estaño para soldadura	100g	0.0496 €/g	4.96 €
	Resistencias serie E12	13 u.	0.012 €/u	0.156 €
	Diodos	3 u.	0.32 €/u	0.96 €
	Optoacoplador LTV-847	3 u.	0.94 €/u	2.82 €
Humanos	Estudios previos	50 h	32 €/h	1600 €
	Diseño e implementación	45 h	32 €/h	1440 €
	Pruebas	30 h	32 €/h	960 €
<b>COSTE CAPÍTULO 01:</b>				<b>4068.55 €</b>

Las horas invertidas en los estudios previos se han invertido en la adquisición de los conocimientos pertinentes como para llevar a cabo el diseño de forma clara y funcional, así como a la comprensión del funcionamiento de cada una de las partes del circuito y comunicación entre las diferentes partes del mismo. En referente a la remuneración por horas se ha tenido en cuenta el coste medio por hora aproximado de un Graduado en Ingeniería en Tecnologías Industriales con 1 año de experiencia.

En el capítulo 2 nos centraremos en estudiar los costes económicos asociados al equipamiento utilizado en el laboratorio para realizar los ensayos, así como de los costes asociados a la realización de las pruebas y ensayos con el equipo.

<b>CAPÍTULO 02 EQUIPAMIENTO Y SOFTWARE AUXILIAR</b>				
RECURSOS	CONCEPTO	CANTIDAD	PRECIO UNITARIO	COSTE
Materiales	Osciloscopio	3 mes.	19,45 €/mes	58.35 €
	Inversor STVAL IHM28	1 u.	467 €/u	467 €
	Fuente de alimentación 30V 3A	3 mes.	12,3 €/mes	36.9 €
	Montaje Servopack + Pareja de motores	3 mes.	119 €/mes	357 €
	Software Matlab (licencia universitaria)	1 u.	0 €/u	0 €
	Software EasyEda (diseño PCB-Esquemáticos).	1 u.	0 €/u	0 €
	Software LT-Spice (simulación de circuitos)	1 u.	0 €/u	0 €
	Software Arduino IDE	1 u.	0 €/m	0 €
	Software Visual Studio Code	1 u.	0 €/u	0 €
Humanos	Estudios previos	30 h.	32 €/h	960 €
	Diseño e implementación	35 h.	32 €/h	1120 €
	Pruebas	30 h.	32 €/h	960€
	Supervisión y dirección	19 h.	103 €/h	1957 €
<b>COSTE CAPÍTULO 02:</b>				<b>5916.25 €</b>

En este capítulo las horas dedicadas a los estudios previos se han invertido en el estudio del funcionamiento del osciloscopio y de las pruebas que se han necesitado realizar con el mismo, así como en la simulación de los circuitos que implementa la tarjeta y el diseño de la PCB para su posible fabricación y esquemático asociado a la misma. Las pruebas realizadas consistieron en ensayos con medidas de tensión a través del osciloscopio con ayuda de un USB y a través del propio visualizador de datos que la aplicación de Arduino IDE implementa. En cuanto a los recursos humanos se ha tenido en cuenta, de igual manera que en capítulo 1, el sueldo medio



de un ingeniero junior y un 20% del tiempo dedicado por el ingeniero junior dedicado además por un ingeniero senior que comprueba y confirma los cálculos y procedimientos a llevar a cabo.

CAPÍTULO 03 MEMORIA				
RECURSOS	CONCEPTO	CANTIDAD	PRECIO UNITARIO	COSTE
Materiales	Ordenador portátil personal	4 meses	34.9 €/mes	139.6 €
Humanos	Redacción de la memoria	90 h	32 €/h	2880 €
<b>COSTE CAPÍTULO 03:</b>				<b>1669.6 €</b>

Por último, se expone a continuación una tabla en la que se resume el coste asociado a cada uno de los capítulos que componen el presupuesto. Además, se expone el coste total asociado al presente proyecto.

RESUMEN PRESUPUESTO	
CONCEPTO	COSTE
Capítulo 01. Tarjeta de control	4068.55 €
Capítulo 02. Equipamiento y software auxiliar	5916.25 €
Capítulo 03. Memoria	1669.6 €
<b>Presupuesto de ejecución material</b>	<b>11654.4 €</b>
Gastos generales (12%)	1398.53 €
Beneficio industrial (6%)	699.27 €
<b>Presupuesto de ejecución por contrata</b>	<b>13752.2 €</b>
IVA (21%)	2887.96 €
<b>Presupuesto base de licitación</b>	<b>16640.16 €</b>

El presupuesto total asciende a la cantidad de **16640.16 €** “DIECISEIS MIL SEISCIENTOS CUARENTA EUROS CON DIECISEIS CÉNTIMOS DE EURO”







UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIERÍA  
INDUSTRIAL VALENCIA

# DISEÑO DE UN SISTEMA DE CONTROL DE INVERSOR PARA UN SERVO SENOIDAL DE 200W

---

## PLANOS

---

Trabajo Final de Grado

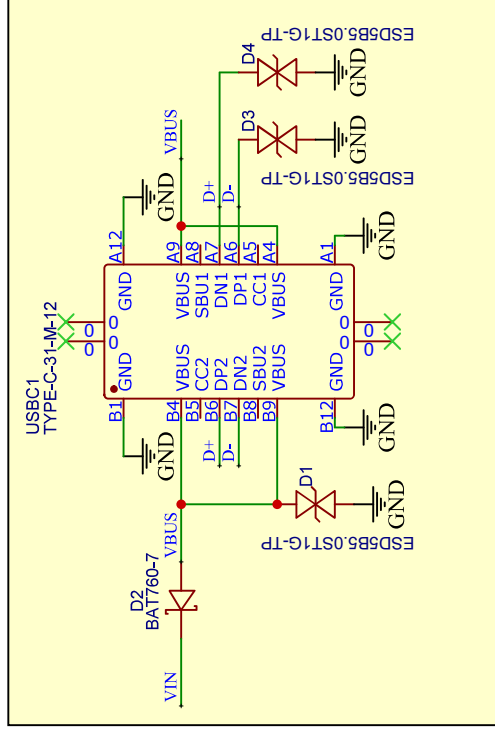
Antonio Poveda Serrano

Julio, 2022

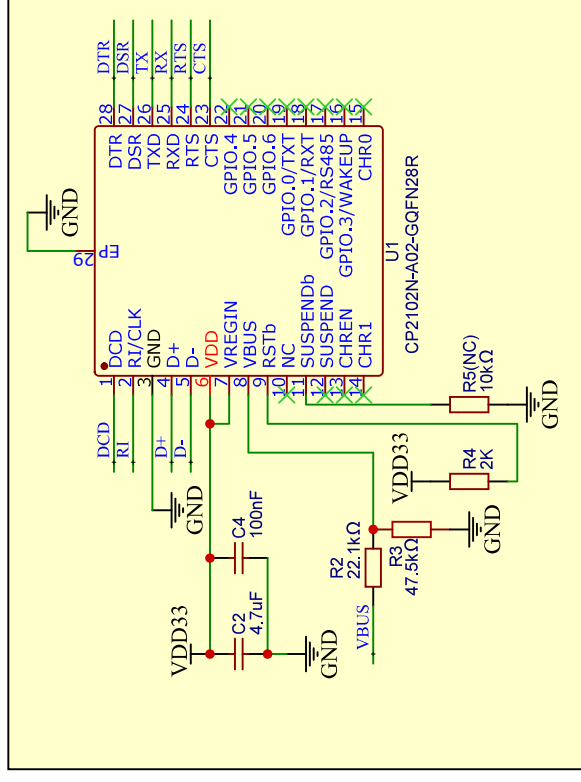
## **ÍNDICE DE LOS PLANOS**

1. Esquemático microcontrolador ESP32 .....	1
2. Esquemáticos de optoacopladores, puertas lógicas y sensores de corriente .....	2
3. Esquemáticos de alimentación y aislamiento USB .....	3
4. Esquemáticos de conectores .....	4
5. Tarjeta PCB .....	5

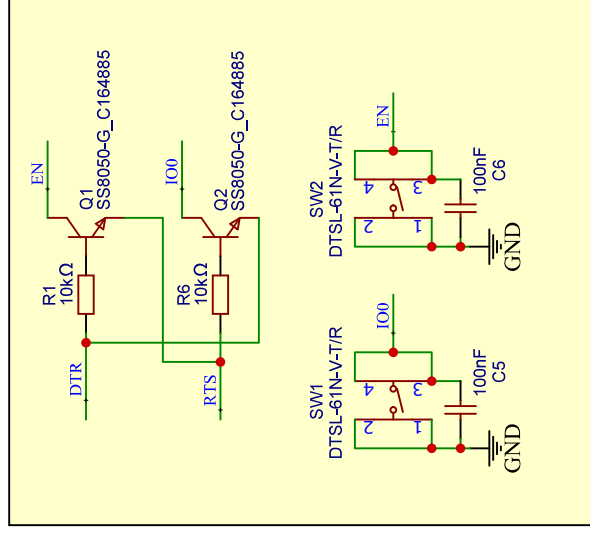
### CONECTOR USB-C



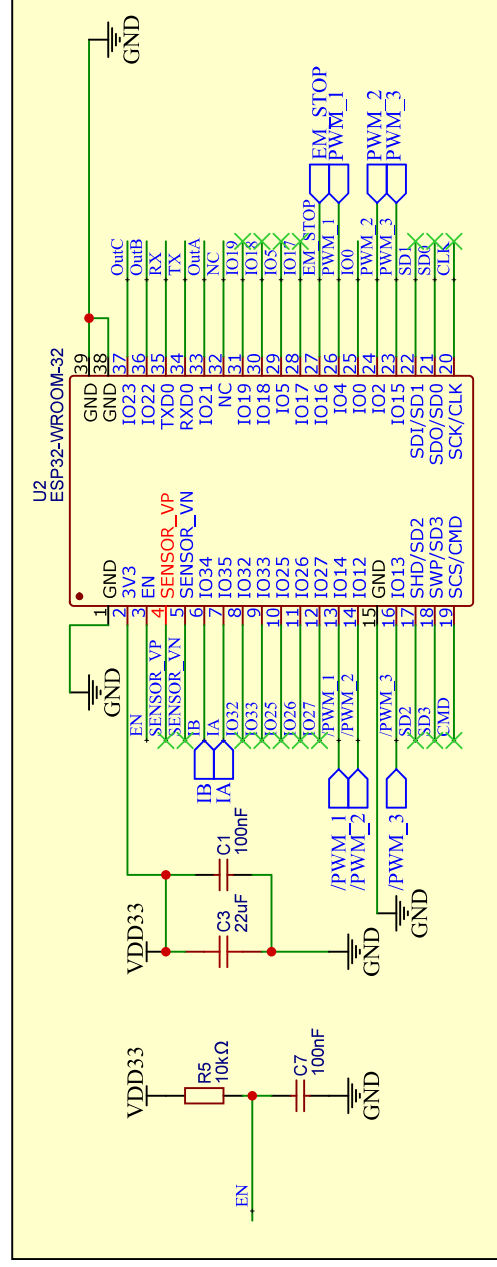
### ADAPTADOR USB - SERIAL



### PULSADORES BOOT Y EN



### MICROCONTROLADOR ESP32



Título

Microcontrolador

REV: 1.0



UPV ETSII

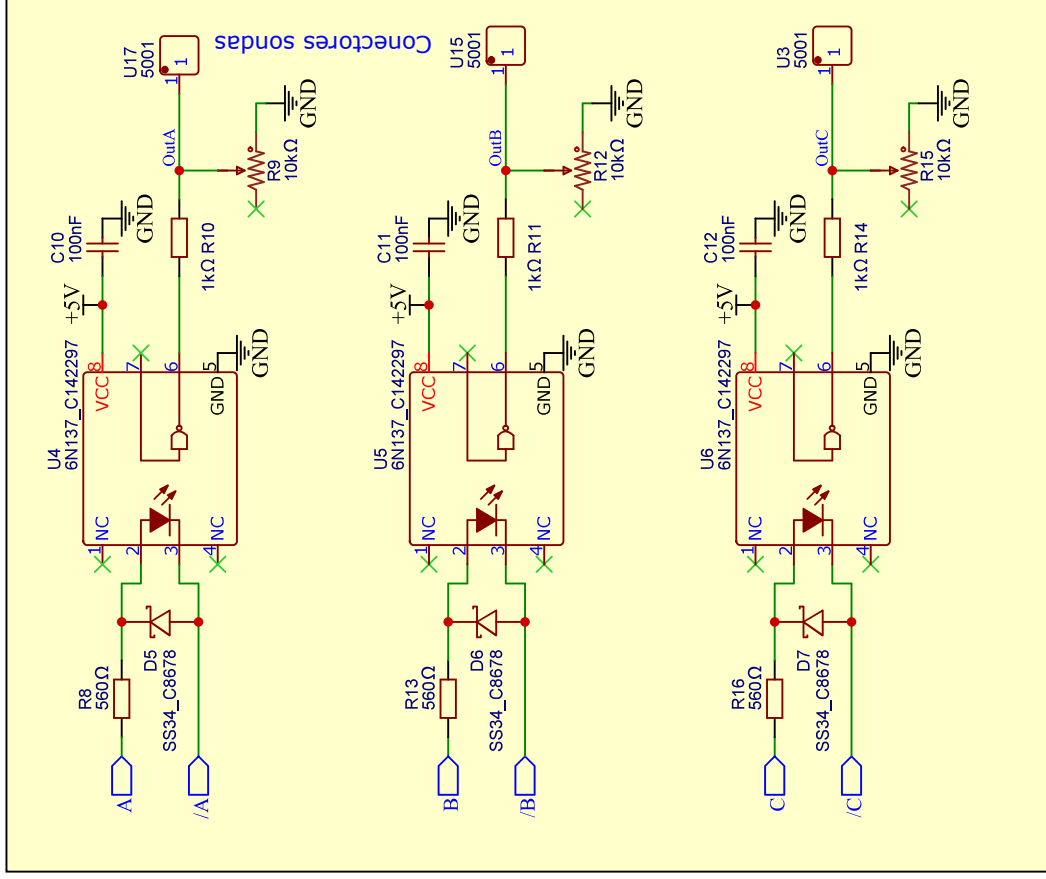
Página:1

Fecha: 2022-04-27

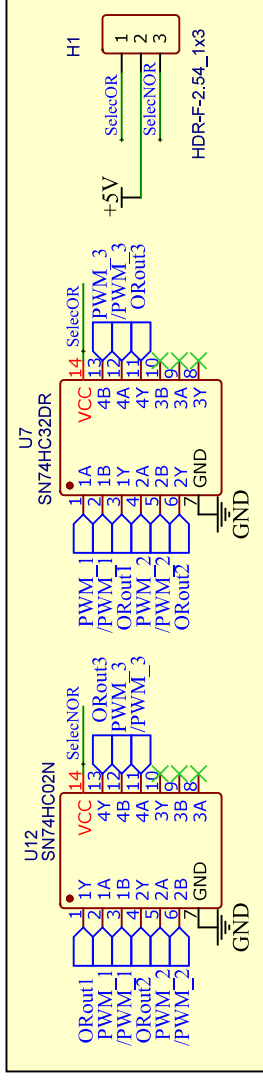
Autor

Antonio Poveda

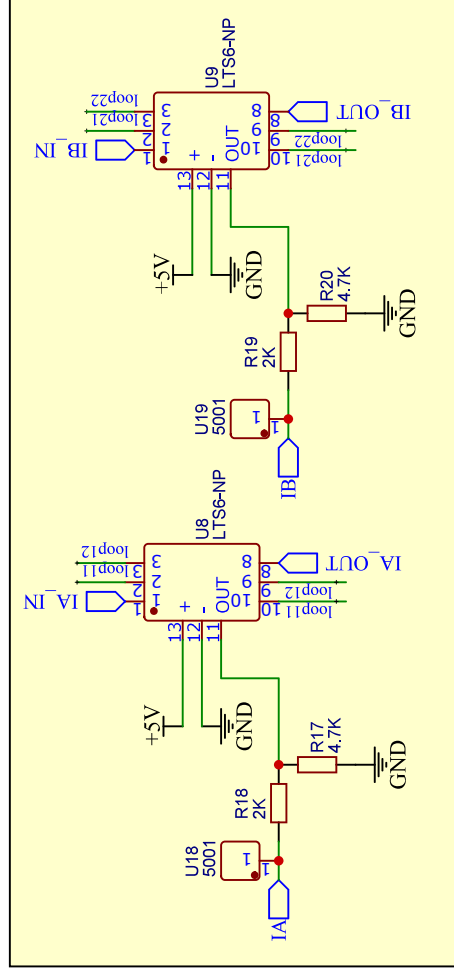
## OPTOACOPLADORES



## PUERTAS L GICAS



## SENSORES DE CORRIENTE



Título

Optoacopladores + puertas lógicas + sensores de corriente

REV: 1.0

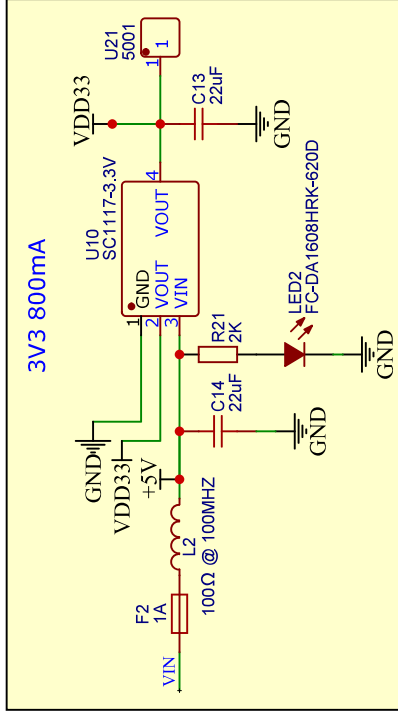


UPV ETSII

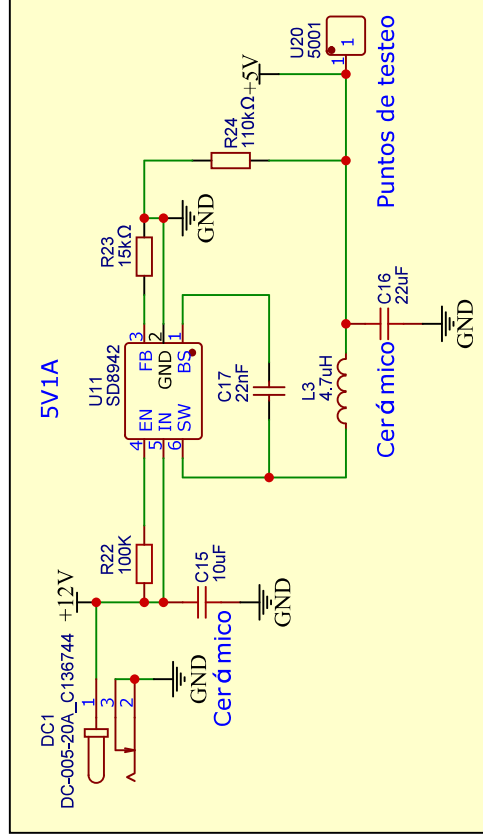
Fecha: 2022-05-02 Autor

Antonio Poveda

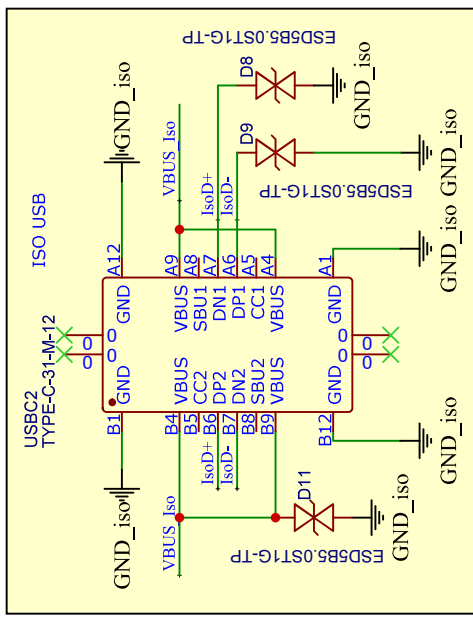
### REGULADOR 5V A 3.3V



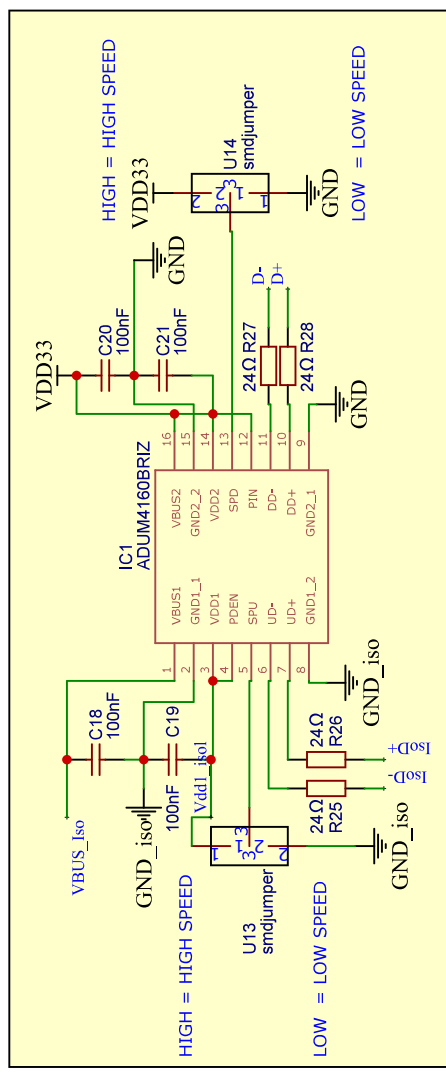
### REGULADOR BUCK 12V A 5V



### CONECTOR USB-C AISLADO



### AISLADOR USB



Título

Alimentación + Aislamiento USB

REV: 1.0



UPV ETSII

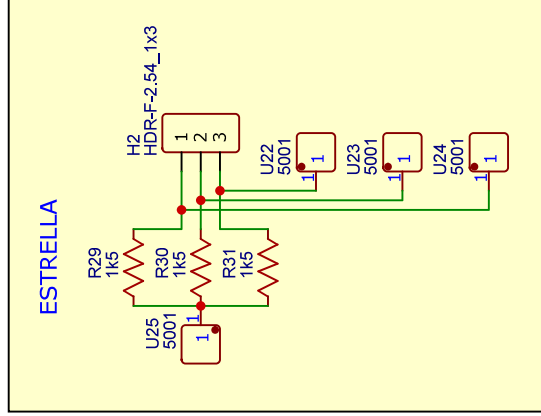
Página:3

Fecha: 2022-05-02 Autor

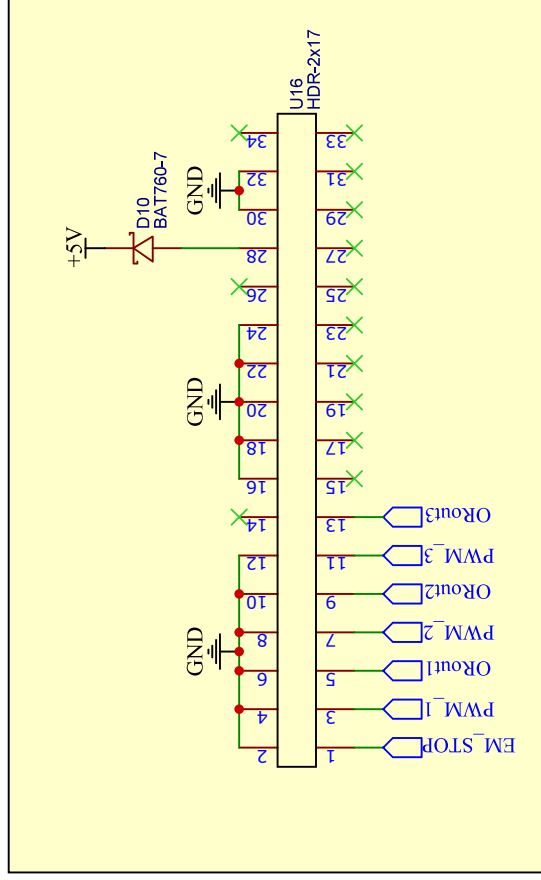
Antonio Poveda



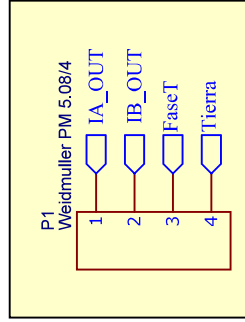
### CONECTOR PARA PRUEBAS DE TENSION DE FASE



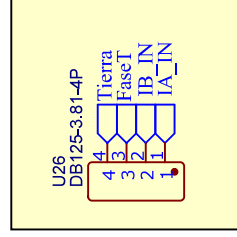
### CONECTOR INVERSOR



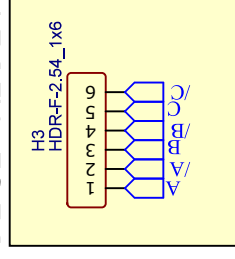
### CONECTOR DE SALIDA DE POTENCIA



### CONECTOR DE ENTRADA DE POTENCIA



### CONECTOR DE ENTRADA DE SEÑAL DE POSICION



Título

Conectores

REV: 1.0

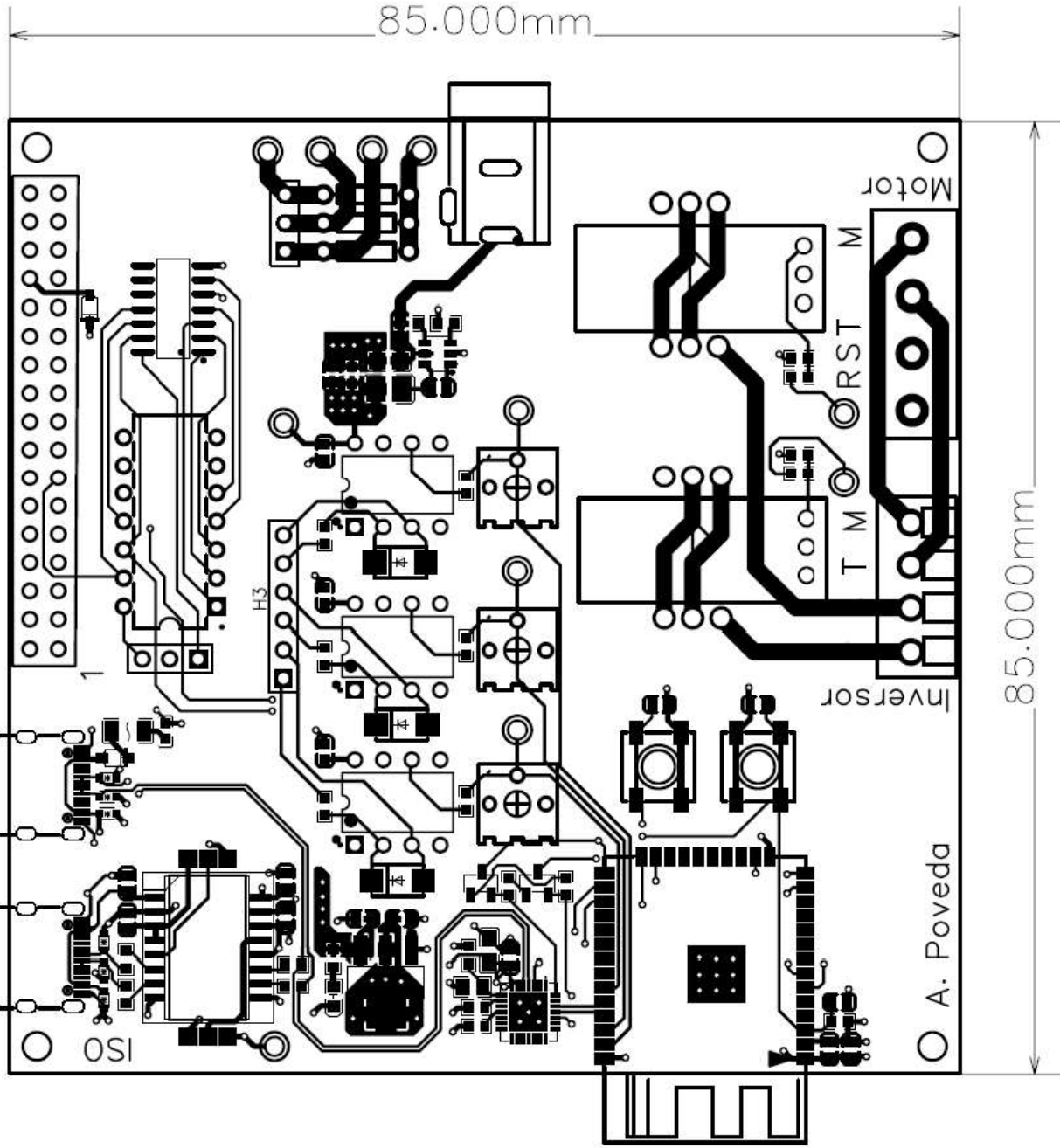


UPV ETSII

Página:4

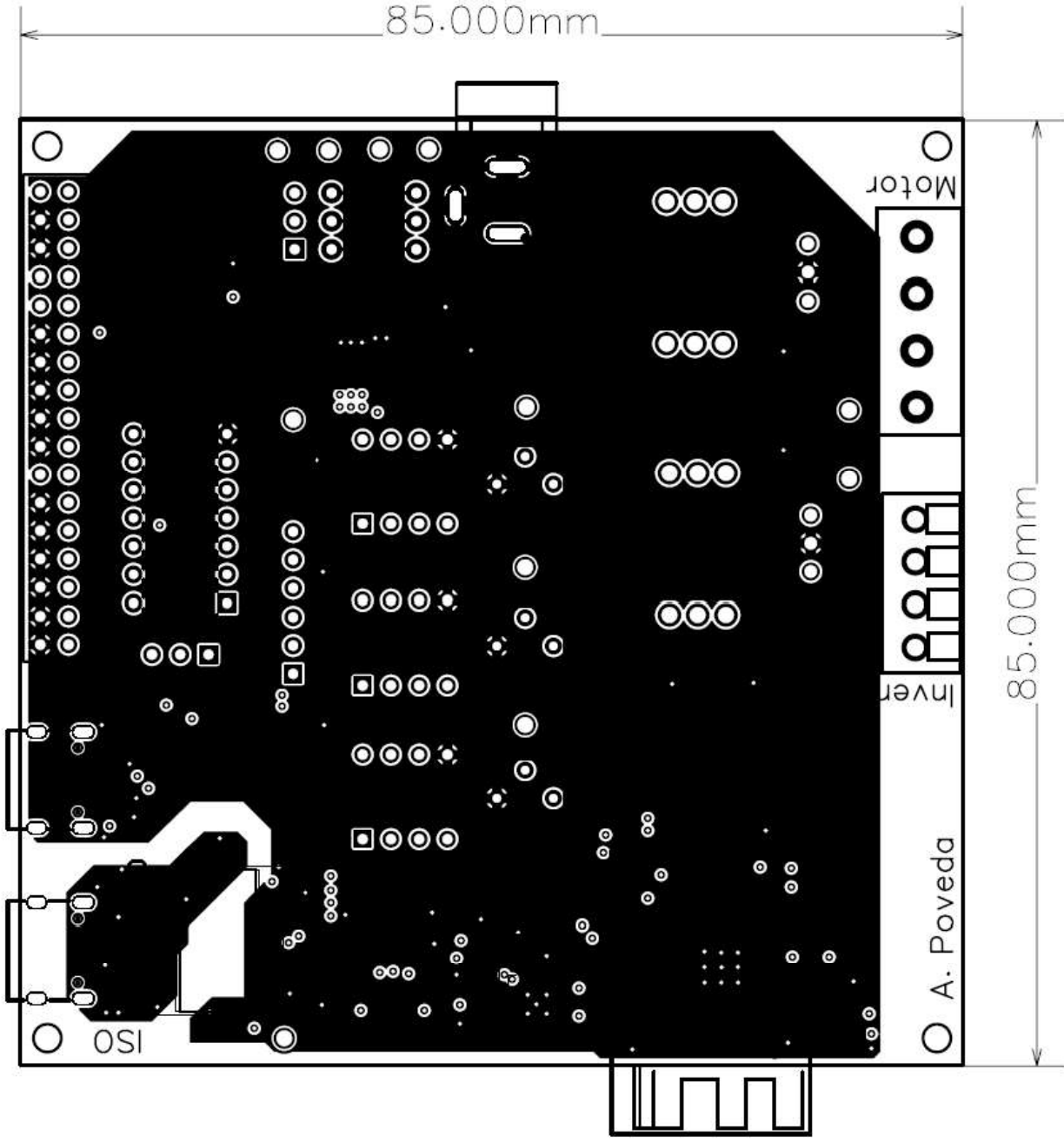
Fecha: 2022-05-11 Autor

Antonio Poveda

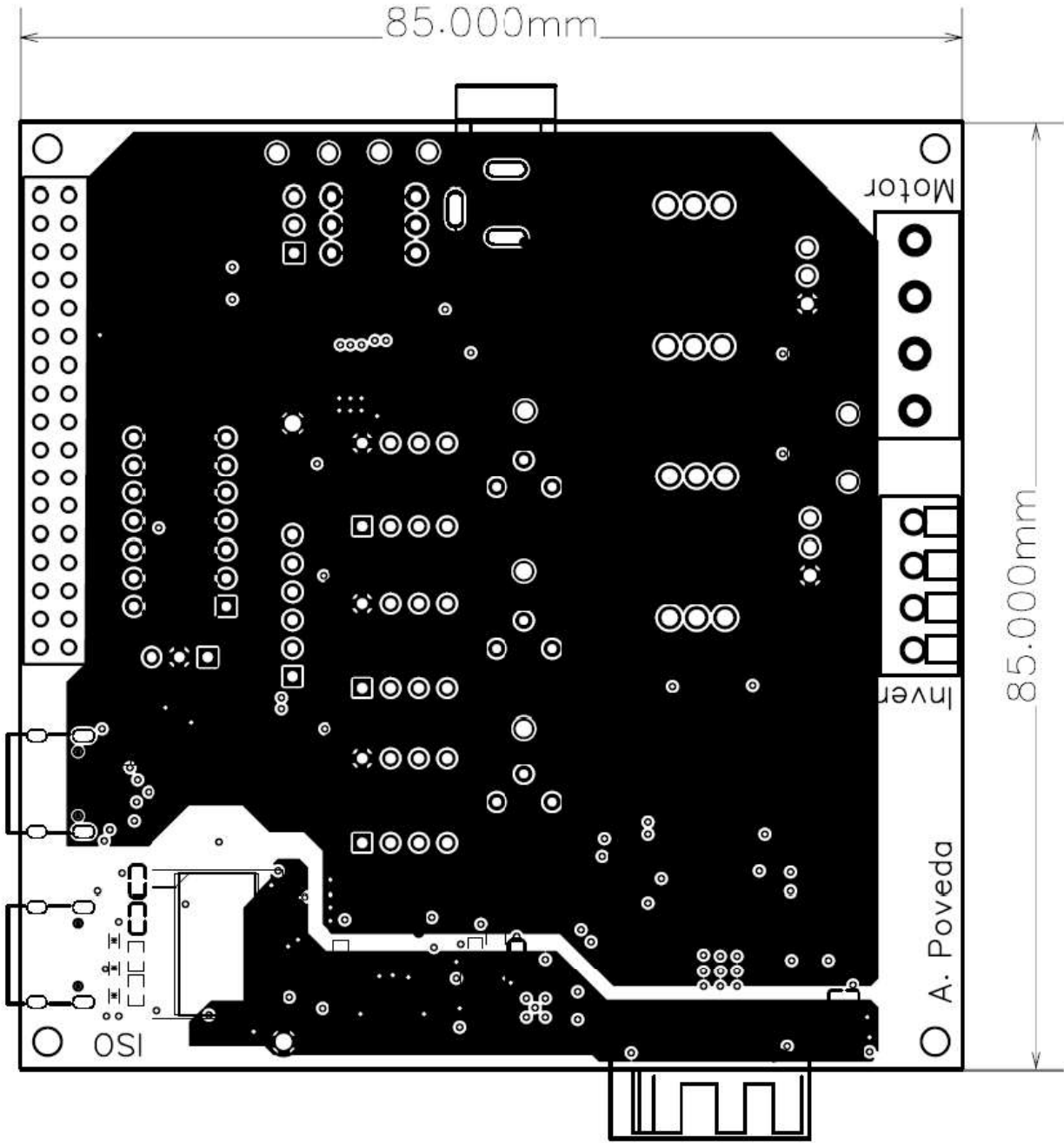


Orificios  
4 x 2.5Ømm

Título	Capa 1	Escala: 2:1
UNIVERSITAT POLITÈCNICA DE VALÈNCIA	UPV ETSII	Pàgina:5
Fecha: 2022-07-02	Autor: Antonio Poveda	



Título	Capa 2	Escala: 2:1
UPV ETSII	UPV ETSII	Página:6
UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Fecha: 2022-07-02	Autor: Antonio Poveda



Título

Capa 3

Escala: 2:1

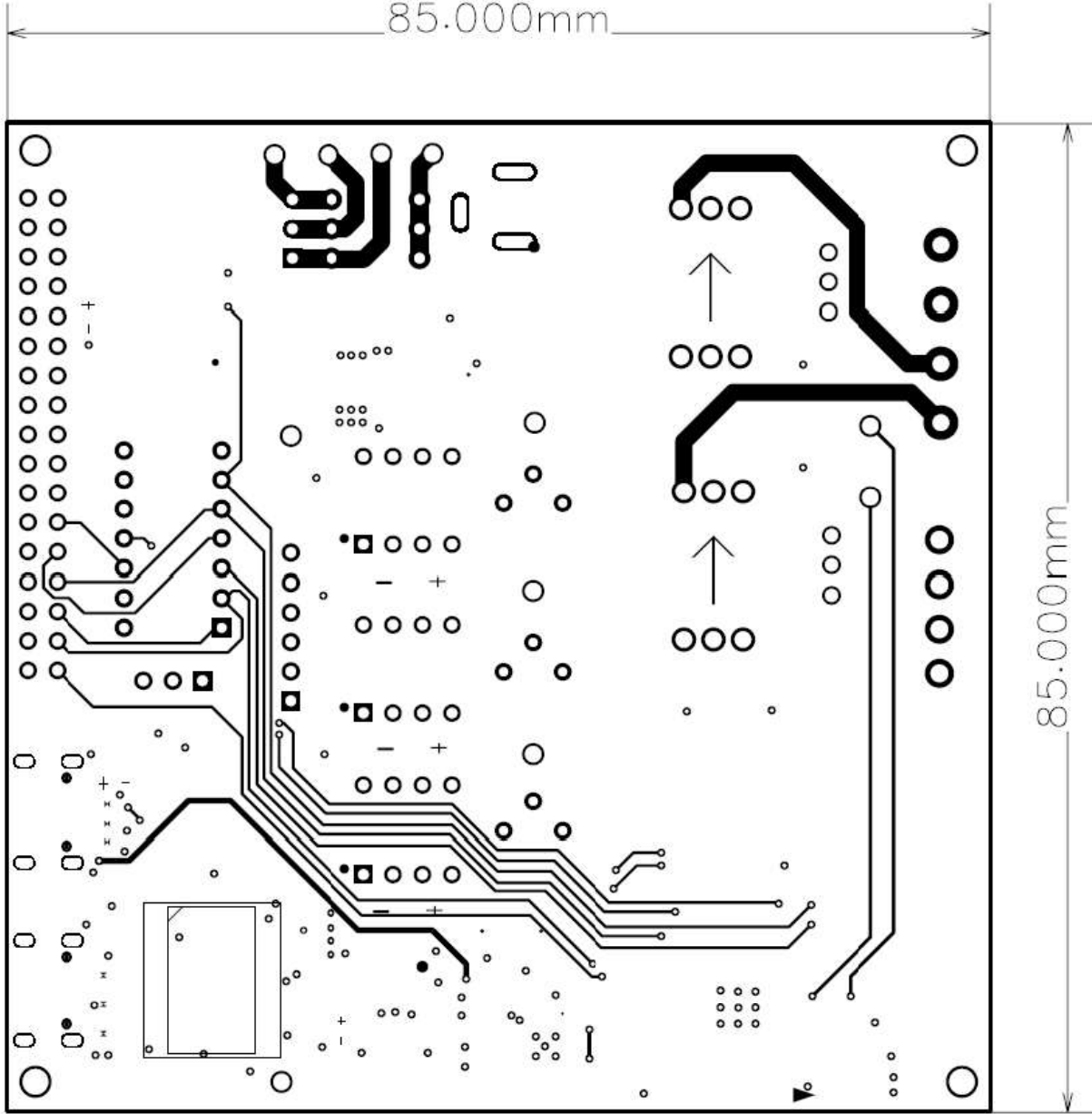


UPV ETSII

Página: 7

Fecha: 2022-07-02

Autor: Antonio Poveda



Orificios  
4 x 2.50mm

Título

Capa 4

Escala: 2:1



UPV ETSII

Página:8

Fecha: 2022-07-02 Autor: Antonio Poveda